

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО

«Сибирский федеральный университет»

институт

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Е.Н. Скуратенко

подпись инициалы, фамилия

« ____ » июня 2017 г

БАКАЛАВРСКАЯ РАБОТА

09.03.03 – Прикладная информатика

код – наименование направления

Разработка информационной системы аттестационного тестирования
медицинских работников

тема

Руководитель

доцент, канд. пед. наук

И. В. Янченко

подпись, дата должность, ученая степень инициалы, фамилия

Выпускник

подпись, дата

Е. Е. Султрекова

инициалы, фамилия

Абакан 2017

Продолжение титульного листа БР по теме «Разработка
информационной системы аттестационного тестирования медицинских
работников»

Консультанты по
разделам:

Аналитический
наименование раздела

подпись, дата

А. В. Ивашина
инициалы, фамилия

Проектный
наименование раздела

подпись, дата

А. В. Ивашина
инициалы, фамилия

Экономический
наименование раздела

подпись, дата

Е. Н. Скуратенко
инициалы, фамилия

Нормоконтролер

подпись, дата

В. И. Кокова
инициалы, фамилия

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Хакасский технический институт – филиал ФГАОУ ВО
«Сибирский федеральный университет»
институт

Прикладная информатика, математика и естественнонаучные дисциплины
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Е.Н.Скуратенко
подпись инициалы, фамилия
« ____ » июня 2017 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

бакалаврской работы, дипломного проекта, дипломной работы, магистерской диссертации

Студенту Султрековой Елене Евгеньевне

фамилия, имя, отчество

Группа 53-1 (ХБ 13-04) Направление (специальность) 09.03.03

номер

код

«Прикладная информатика»

наименование

Тема выпускной квалификационной работы «Разработка информационной Системы аттестационного тестирования медицинских работников»

Утверждена приказом по институту № 156 от 28.02.2017 г.

Руководитель ВКР И. В. Янченко, доцент кафедры ПИМиЕД, канд. пед. наук, ХТИ – филиал СФУ

инициалы, фамилия, должность, ученое звание и место работы

Исходные данные для ВКР – заказ учреждения ГКУЗ РХ «РМИАЦ»

Перечень разделов ВКР

1. Аналитический раздел «Теоретический анализ процесса аттестации медицинских работников Республики Хакасия».

2. Проектный раздел «Практическая реализация разработки информационной системы аттестационного тестирования медицинских работников (модули администратора и модератора)».

3. Экономический раздел «Оценка экономической эффективности внедрения информационной системы аттестационного тестирования медицинских работников в процесс аттестации».

Перечень графического материала –

Руководитель ВКР

подпись

И. В. Янченко

инициалы и фамилия

Задание принял к исполнению

Е. Е. Султрекова

подпись, инициалы и фамилия студента

« 28 » февраля 2017 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка информационной системы аттестационного тестирования медицинских работников» содержит // страниц текстового документа, // рисунка, // таблиц, // формулы, // приложений, // использованный источник.

СИСТЕМА АТТЕСТАЦИОННОГО ТЕСТИРОВАНИЯ, АТТЕСТАЦИЯ МЕДИЦИНСКИХ РАБОТНИКОВ, ПРОЦЕСС АТТЕСТАЦИИ, ТЕСТИРОВАНИЕ, КЛИЕНТ-СЕРВЕР, МОДУЛЬ АДМИНИСТРАТОРА, МОДУЛЬ МОДЕРАТОРА, БАЗА ДАННЫХ, СУБД MYSQL, PHP, HTML, JAVASCRIPT, CSS.

Объектом исследования является процесс аттестационного тестирования медицинских работников.

Целью ВКР является разработка и внедрение системы аттестационного тестирования медицинских работников в республике Хакасия.

Основными задачами ВКР являются анализ информации о процессе аттестации медицинских работников, проводимом Министерством здравоохранения Республики Хакасия; проектирование функциональной модели аттестации медицинских работников; анализ существующих систем тестирования и обоснование актуальности разработки информационной системы тестирования медицинских работников своими средствами; разработка информационной системы аттестационного тестирования медицинских работников (модули администратора и модератора) и оценка экономической эффективности внедрения системы.

В результате реализации цели и задач ВКР разработана и внедрена система аттестационного тестирования медицинских работников на основе архитектуры «клиент-сервер».

Разработанная система позволяет автоматизировать процесс аттестационного тестирования медицинских работников на уровне республики Хакасия.

ZUSAMMENFASSUNG

Das Thema der Diplomarbeit ist «Die Entwicklung des Informationssystems des Qualifikationstests der medizinische Arbeiter». Die Arbeit hat ____ Seiten, ____ Illustrationen, ____ Tabellen, ____ Formeln, ____ Anlagen, ____ Literaturquellen.

Das System des Qualifikationstests, der Qualifikation der medizinische Arbeiter, die Testen, der Kunde-Server, das Administratormodul, das Moderatormodul, die Datenbasis, das DBMS MySQL, PHP, HTML, JAVASCRIPT, CSS.

Das Forschungsobjekt ist der Prozess des Qualifikationstests der medizinische Arbeiter.

Das Ziel. Die Entwicklung und die Einführung des Informationssystems des Qualifikationstests der medizinische Arbeiter in der Republik Chakassien.

Die Aufgaben. Die Analyse der Information über den Qualifikationsprozess der medizinische Arbeiter durch das Ministerium für Gesundheit der Republik Chakassien durchgeführt, die Projektion des Funktionsmodell des Qualifikation der medizinische Arbeiter, die Analyse der bestehenden Testsysteme und die Erhärtung der Aktualität der Entwicklung des Informationssystems des Qualifikationstests der medizinische Arbeiter ihre Mittel des Web-Programmierens, die Entwicklung des Informationssystems des Qualifikationstests der medizinische Arbeiter (das Administratormodul und das Moderatormodul) und die Einschätzung die ökonomisch Effektivität der Einführung des Systems.

Infolge der Realisierung der Ziele und der Aufgaben der Diplomarbeit ist das System des Qualifikationstests der medizinische Arbeiter auf dem Gelände der Client-Server-Architektur entwickelt und einführt.

Das entwickelte System erlaubt der Prozess des Qualifikationstests der medizinische Arbeiter in der Republik Chakassien automatisieren.

Der Leiter im Deutschen

_____ E. A. Nikitina

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	9
1 Теоретический анализ процесса аттестации медицинских работников Республики Хакасия.....	12
1.1 Организационная характеристика Министерства здравоохранения Республики Хакасия как уполномоченного присваивать квалификационные категории медицинскими и фармацевтическими работникам	12
1.2 Требования к порядку и проведению процесса аттестации медицинских и фармацевтических работников	16
1.3 Программные продукты, имеющиеся в распоряжении органа управления здравоохранением.....	18
1.4 Модель процесса аттестации медицинских работников.....	21
1.5 Постановка цели и задач проектирования	27
1.6 Сравнительный анализ существующих систем	28
1.7 Обоснование выбора средств разработки информационной системы аттестационного тестирования	31
1.7.1 Выбор информационного обеспечения и его обоснование	32
1.7.2 Обоснование выбора языка программирования для разработки информационной системы.....	34
Вывод по аналитическому разделу.....	39
2 Практическая реализация разработки информационной системы аттестационного тестирования медицинских работников (модули администратора и модератора)	41
2.1 Проектирование модуля администратора и модератора системы аттестационного тестирования медицинских работников.....	41
2.1.1 Определение и анализ данных, необходимых для выполнения каждой функции процесса	41

2.1.2 Проектирование модели структуры данных	46
2.2 Программно-техническое обеспечение разработки модуля администратора и модератора.....	52
2.2.1 Файловая структура модуля администратора и модератора	52
2.2.2 Структура страниц модуля администратора и модератора	54
2.2.3 Описание модуля администратора	57
2.2.4 Описание модуля модератора	67
2.2.5 Результат работы модулей администратора и модератора	68
2.2.6 Описание работы системы аттестационного тестирования.....	72
2.2.7 Обеспечение информационной безопасности.....	73
Вывод по проектному разделу	75
3 Оценка экономической эффективности внедрения информационной системы аттестационного тестирования медицинских работников в процесс аттестации	76
3.1 Расчет эксплуатационных затрат.....	76
3.2 Оценка эффективности проектов в зависимости от технических, технологических, отраслевых или региональных особенностей	80
3.3 Оценка рисков проекта разработки системы аттестационного тестирования медицинских работников	82
Вывод по экономическому разделу.....	84
ЗАКЛЮЧЕНИЕ	86
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	88
ПРИЛОЖЕНИЕ А-У	91-125

ВВЕДЕНИЕ

Аттестация является основной формой комплексной оценки персонала в здравоохранении, по результатам которой принимается решение о соответствии занимаемой должности и служебно-профессиональном продвижении, сокращении штата, присваиваются квалификационные категории (Меркулов М.М., 2004). Таким образом, аттестацию можно рассматривать как один из механизмов государственного контроля за качеством подготовки специалистов, а, следовательно – и качеством оказания медицинской помощи населению, эффективностью деятельности медицинской организации [4]. Возможность повышения категории после процесса аттестации медицинского работника позволяет максимально раскрыть потенциал личности и соответственно улучшить качество предоставляемых им медицинских услуг.

Вследствие того, что медицинские технологии стремительно развиваются, многократно возросли требования к уровню подготовки и квалификации специалистов государственных и муниципальных учреждений здравоохранения, способных работать на современном высокотехнологичном медицинском оборудовании. Поэтому аттестацию медицинских работников не следует рассматривать в отрыве от процесса сертификации: аттестация является одной из форм повышения квалификации специалистов, однако, это две разные процедуры [4].

В настоящее время аттестационное тестирование медицинских и фармацевтических работников для получения квалификационной категории осуществляется с помощью системы «Система тестирования медицинских работников 2.1», которая разработана отделом автоматизированных систем управления Государственного казённого учреждения здравоохранения Республики Хакасия «Республиканский медицинский информационно-аналитический центр» (ГКУЗ РХ «РМИАЦ»).

Функционал системы не соответствует современным требованиям, в процессе использования системы аттестационного тестирования появились новые задачи и выявились критические ошибки. Таким образом, актуальность выбранной темы выпускной квалификационной работы (ВКР) связана с необходимостью внедрения новой информационной системы аттестационного тестирования медицинских работников на уровне республики Хакасия.

Целью выпускной квалификационной работы является разработка и тестирование системы аттестационного тестирования медицинских работников республики Хакасия.

В соответствии с поставленной целью объектом исследования является процесс аттестационного тестирования медицинских работников.

Предмет исследования – информационная система аттестационного тестирования медицинских работников республики Хакасия.

Для достижения поставленной цели необходимо решить следующие задачи:

1. провести сбор и анализ информации о процессе аттестации медицинских работников Республики Хакасия;
2. построить функциональную модель аттестации медицинских работников;
3. определить цель и задачи проектирования системы аттестационного тестирования;
4. провести анализ существующих систем тестирования и обосновать актуальность разработки информационной системы тестирования медицинских работников своими средствами;
5. провести теоретический анализ и выбрать программные средства разработки информационной системы тестирования медицинских работников;
6. разработать и провести тестирование информационной системы;

7. оценить экономическую эффективность внедрения информационной системы аттестационного тестирования медицинских работников в процесс аттестации.

Для достижения поставленной цели и решения задач ВКР были использованы теоретические и эмпирические методы исследования.

В связи со значительным объемом работ по созданию информационной системы аттестационного тестирования медицинских работников в условиях ограниченности временных ресурсов система разрабатывалась модульно совместно со студентом группы 53-1 Дресвянкиным Вадимом Сергеевичем.

В проектной части данной ВКР представлено описание и реализация модуля администратора и модератора системы аттестационного тестирования медицинских работников, в работе Дресвянкина Вадима Сергеевича представлен модуль тестирования медицинских работников.

1 Теоретический анализ процесса аттестации медицинских работников Республики Хакасия

В теоретическом разделе ВКР проанализирован процесс аттестации медицинских работников Республики Хакасия, на основании которого спроектирована функциональная модель аттестации медицинских работников. Для достижения поставленной цели проектирования проанализированы рынок существующих систем тестирования и программные средства разработки информационной системы тестирования медицинских работников.

1.1 Организационная характеристика Министерства здравоохранения Республики Хакасия как уполномоченного присваивать квалификационные категории медицинскими и фармацевтическими работникам

Исполнительным органом государственной власти республики Хакасия, осуществляющим функции по реализации государственной политики и управлению в сфере здравоохранения, включая вопросы организации оказания медицинской помощи и лекарственного обеспечения на основании Положения «О Министерстве здравоохранения республики Хакасия», утвержденном Постановлением Правительства Республики Хакасия от 11 июня 2009 г. № 260 является Министерство здравоохранения республики Хакасия.

Краткое наименование – Минздрав Хакасии. Юридический адрес учреждения: 655017, республика Хакасия, город Абакан, улица Крылова, 72 (3 этаж). Контактные телефоны: секретарь – (3902) 29-50-91, факс (3902) 23-27-25, электронный адрес: mz@r-19.ru.

Министерство здравоохранения республики Хакасия управляет 11 больницами, 4 диспансерами, 1 общественной организацией, 6

поликлиниками, 8 районными больницами, 3 станциями скорой медицинской помощи, 6 учреждениями детства и родовспоможения и 6 учреждений особого типа. Также в 2015 году в республике Хакасия численность врачей составляла 1792 человек, среднего медицинского персонала – 5055 человек.

Министерство здравоохранения республики Хакасия возглавляет министр, назначаемый на должность (освобождаемый от должности) Главой республики Хакасия – Председателем Правительства республики Хакасия.

Министр несет персональную ответственность за выполнение возложенных на Министерство здравоохранения полномочий.

Министр имеет заместителей, в том числе первого заместителя, назначаемых на должность и освобождаемых от должности министром по согласованию с Главой республики Хакасия – Председателем Правительства республики Хакасия [1].

Структурная схема Министерства здравоохранения Республики Хакасия представлена на рисунке 1.



Рисунок 1 – Структура Министерства здравоохранения Республики Хакасия

Основными задачами Министерства здравоохранения Республики Хакасия являются:

1. обеспечение реализации конституционных прав граждан Российской Федерации на гарантированное получение медицинской помощи, лекарственных средств, изделий медицинского назначения;
2. реализация государственной политики и государственного управления в области здравоохранения;
3. обеспечение населения своевременной, доступной, качественной медицинской помощью [1].

Министерство здравоохранения Республики Хакасия осуществляет следующие полномочия:

1. осуществление закупок товаров, работ, услуг для обеспечения государственных нужд в сфере здравоохранения в соответствии с законодательством Российской Федерации о контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд;
2. развитие информационной системы в сфере здравоохранения;
3. защиту сведений, составляющих государственную тайну, и иной информации в соответствии с действующим законодательством;
4. контролирует санитарно-эпидемиологическое благополучие населения;
5. развитие медицинских организаций, оказывающих медицинскую помощь детям, с учетом обеспечения благоприятных условий для пребывания в них детей, в том числе детей-инвалидов, и возможности пребывания с ними родителей и (или) иных членов семьи;
6. реализацию мер социальной поддержки специалистам с медицинским образованием и лицам, обучающимся на основании договора о целевом обучении для последующего трудоустройства в медицинские организации государственной системы здравоохранения Республики Хакасия.

7. создание территориальных аттестационных комиссий для получения квалификационных категорий медицинскими работниками и фармацевтическими работниками [1].

Министерство здравоохранения является учредителем ГКУЗ РХ «РМИАЦ», которое создано в соответствии с Постановлением Правительства Республики Хакасия от 30.08.2007 № 274 для выполнения функций по обеспечению реализации предусмотренных законодательством Российской Федерации полномочий в сфере организации сбора, обработки и анализа медицинских статистических данных о деятельности учреждений здравоохранения, состоянии здоровья населения, управления системой медицинского статистического учета и отчетности в организациях и учреждениях здравоохранения Республики Хакасия [21].

Основными целями деятельности ГКУЗ РХ «РМИАЦ» являются:

1. организационно-методическое руководство по формированию единой информационной системы здравоохранения в Республике Хакасия, созданию и сопровождению автоматизированных систем управления здравоохранением Республики Хакасия;
2. координация деятельности службы медицинской статистики и медицинского статистического обеспечения органов управления здравоохранением Республики Хакасия;
3. анализ медико-статистической информации о состоянии здоровья населения и деятельности здравоохранения в Республике Хакасия;
4. внедрение новых технологий сбора и обработки медико-статистической информации;
5. обеспечение безопасности автоматизированных информационных систем здравоохранения Республики Хакасия;
6. обеспечение безопасности автоматизированных информационных систем здравоохранения Республики Хакасия;
7. формирование единой информационной системы здравоохранения Российской Федерации на базе современных компьютерных технологий

межотраслевой системы сбора, обработки, хранения и предоставления информации, обеспечивающей динамическую оценку состояния здоровья и информационную поддержку принятия решений, направленных на его улучшение [21].

1.2 Требования к порядку и проведению процесса аттестации медицинских и фармацевтических работников

В соответствии со статьей 72 Федерального закона от 21.11.2011 № 323-ФЗ "Об основах охраны здоровья граждан в Российской Федерации" медицинские и фармацевтические работники имеют право на прохождение аттестации для получения квалификационной категории в порядке и в сроки, определяемые уполномоченным федеральным органом исполнительной власти [2]. Согласно Постановлению Правительства Республики Хакасия от 27.03.2014 № 138 аттестация проводится аттестационными комиссиями по трем квалификационным категориям: второй, первой и высшей.

Основные положения документа о Порядке и сроках прохождения медицинскими работниками и фармацевтическими работниками аттестации для получения квалификационной категории об аттестации специалистов для присвоения более высокой квалификационной категории [2]:

1. аттестация специалистов со средним и высшим медицинским и фармацевтическим образованием проводится по специальностям, предусмотренным действующей номенклатурой специальностей специалистов, имеющих медицинское и фармацевтическое образование;

2. аттестация проводится один раз в пять лет, при этом присвоенная квалификационная категория действительна на всей территории Российской Федерации в течение пяти лет со дня издания распорядительного акта о присвоении;

3. специалисты могут претендовать на присвоение более высокой квалификационной категории не ранее чем через три года со дня издания распорядительного акта о присвоении квалификационной категории;

4. при аттестации оцениваются теоретические знания и практические навыки, необходимые для выполнения профессиональных обязанностей по соответствующим специальностям и должностям, на основе результатов квалификационного экзамена;

5. квалификационный экзамен включает в себя экспертную оценку отчета о профессиональной деятельности специалиста, тестовый контроль знаний и собеседование;

6. квалификационные категории, присвоенные специалистам до вступления в силу настоящего Порядка, сохраняются в течение срока, на который они были присвоены;

Специалист для присвоения квалификационной категории должен отвечать соответствующим требованиям, представленным в таблице 1.

Таблица 1 – Требования для присвоения квалификационной категории

Квалификационные категории		
вторая	первая	высшая
Иметь теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности	Иметь теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности и смежных дисциплин	Иметь высокую теоретическую подготовку и практические навыки в области осуществляемой профессиональной деятельности, знать смежные дисциплины
Использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности	Использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности	Использовать современные методы диагностики, профилактики, лечения, реабилитации и владеть лечебно-диагностической техникой в области осуществляемой профессиональной деятельности
Ориентироваться в современной научно-технической информации	Ориентироваться в современной научно-технической информации	Ориентироваться в современной научно-технической информации
Владеть навыками анализа количественных и качественных показателей работы, составления отчета о работе	Уметь квалифицированно провести анализ показателей профессиональной деятельности	Уметь квалифицированно оценить данные специальных методов исследования с целью установления диагноза

Окончание таблицы 1

	Участвовать в решении тактических вопросов организации профессиональной деятельности	Использовать современную научно-техническую информацию для решения тактических и стратегических вопросов профессиональной деятельности;
Иметь стаж работы по специальности (в должности) не менее трех лет	Иметь стаж работы по специальности (в должности) не менее пяти лет	Иметь стаж работы по специальности (в должности) не менее семи лет

1.3 Программные продукты, имеющиеся в распоряжении органа управления здравоохранением

Использование информационных технологий становится обязательным условием совершенствования системы здравоохранения условием, обеспечивающим повышение эффективности организации здравоохранения и улучшение качества оказания медицинской помощи. Информатизация здравоохранения выделена отдельным пунктом в Концепции развития системы здравоохранения Российской Федерации до 2020 года [23].

Основные программные продукты, имеющиеся в распоряжении органа управления здравоохранением Республики Хакасия, представлены в таблице 2.

Таблица 2 – Программные продукты Минздрава Хакасии

№ п/п	Наименование программного продукта	Уровень разработки программы (федеральный/региональный)	Сфера (область) использования	Краткое описание программного продукта	Пользователи программы
1	РИАМС «ПроМед»	Региональный	Деятельность в сфере здравоохранения	Система автоматизации документооборота для лечебно-профилактических учреждений, в которой объединены электронные медицинские карты, данные медицинских исследований, финансовая и административная информация и т.д.	Минздрав Хакасии, медицинские организации

Продолжение таблицы 2

2	Электронная регистратура, Web-сервис к врачу 19.pfk-vrachu.mz19.ru	Региональный	Электронная запись на прием к врачу	Предоставление государственной услуги «Заявка (запись) на прием к врачу» в электронном виде	Минздрав Хакасии, медицинские организации
3	Программный комплекс АСУЛОН	Региональный	Льготное лекарственное обеспечение жителей региона	Выписка рецептов на лекарственные средства и изделия медицинского назначения	Минздрав Хакасии, медицинские организации
4	Федеральный регистр медицинских работников	Федеральный	Мониторинг кадрового состава системы здравоохранения	Сведения о медицинских работниках Республики Хакасия	Минздрав Хакасии, медицинские организации
5	Федеральный регистр медицинских организаций	Федеральный	Мониторинг информации о медицинских организациях	Данные о медицинских организациях и их структурных подразделениях, оказывающих медицинскую помощь	Минздрав Хакасии, медицинские организации
6	Федеральный регистр лиц, инфицированных ВИЧ	Федеральный	Мониторинг данные о пациентах, больных ВИЧ	Сбор, обработка и хранение данных пациентов, ведение карт лечения, диспансерного наблюдения, информация об исследованиях, лекарственных препаратах	Минздрав Хакасии, медицинские организации
7	Федеральный регистр лиц, больных туберкулезом	Федеральный	Мониторинг данных о пациентах, больных туберкулезом	Сбор, хранение, просмотр и анализ данных о пациентах, больных туберкулезом, просмотр форм отчетности.	Минздрав Хакасии, медицинские организации
8	Федеральный регистр онкологических больных	Федеральный	Данные о пациентах, больных онкологическими заболеваниями	Данные о пациентах, больных онкологическими заболеваниями	ГБУЗ РХ «Республиканский клинический онкологический диспансер»
9	Система АХД на базе 1С: Медицина	Региональный	Административно-хозяйственная деятельность	Сбор информации об административно-хозяйственной деятельности медицинских организаций	Медицинские организации
10	Федеральный регистр диспансеризации несовершеннолетних декретированных возрастов	Федеральный	Учет диспансеризации	Ведение регистра диспансеризации несовершеннолетних декретированных возрастов	Медицинские организации

Продолжение таблицы 2

11	Подсистема мониторинга проведения диспансеризации детей-сирот и детей, находящихся в трудной жизненной ситуации	Федеральный	Учет диспансеризации	Диспансеризация детей, находящихся в стационарных учреждениях, детей сирот и детей, оставшихся без попечения родителей	Минздрав Хакасии, медицинские организации
12	Подсистема мониторинга санаторно-курортного лечения	Федеральный	Санаторно-курортное лечение	Направление на санаторно-курортное лечение	Минздрав Хакасии
13	Подсистема мониторинга реализации государственного задания по оказанию высокотехнологичной медицинской помощи за счет средств федерального бюджета	Федеральный	Высокотехнологичная медицинская помощь	Направление граждан на высокотехнологичную медицинскую помощь	Минздрав Хакасии, медицинские организации
14	Федеральный регистр 7 нозологий	Федеральный	Мониторинг сведений о пациентах	Автоматизация процессов сбора, обработки, передачи, хранения, распределённого доступа и поиска лиц, больных гемофилией, муковисцидозом, гипофизарным нанизмом, болезнью Гоше, злокачественными новообразованиями лимфоидной, кроветворной и родственных им тканей, рассеянным склерозом, лиц после трансплантации органов и (или) тканей	Минздрав Хакасии, медицинские организации

Окончание таблицы 2

15	Подсистема ведения Федерального регистра лиц, страдающих жизнеугрожающими и хроническими прогрессирующими редкими (орфанными) заболеваниями	Федеральный	Мониторинг сведений о пациентах	Автоматизация процессов сбора, обработки, передачи, хранения, распределённого доступа и поиска лиц, страдающих жизнеугрожающими и хроническими прогрессирующими редкими (орфанными) заболеваниями	Минздрав Хакасии, медицинские организации
16	Автограф	Региональный	Диспетчеризация санитарного транспорта на основе ГЛОНАСС	Система мониторинга автомобильного транспорта позволяет видеть точное местоположение машины на карте и измерить простой и пробег, выбрать целесообразный маршрут движения	Медицинские организации
17	ИПРА-Веб	Региональный	Ведение карт реабилитации инвалидов	Получение и направление в МСЭ исполненных карт реабилитации инвалидов	Минздрав Хакасии, медицинские организации

Тестовый контроль знаний специалиста осуществляется в системе «Система тестирования медицинских работников 2.1», разработанной и обслуживаемой ГКУЗ РХ «РМИАЦ», в которой были выявлены критические ошибки.

1.4 Модель процесса аттестации медицинских работников

Для определения как организован процесс аттестации медицинских работников с точки зрения участников и каков документооборот происходит в процессе смоделируем этот процесс.

Для моделирования процессов аттестации медицинских работников использовалась методология функционального моделирования IDEF0, позволившая представить модель системы аттестационного тестирования

медицинских работников в виде набора взаимосвязанных функций (функциональных блоков — в терминах IDEF0) и интерфейсных дуг.

Функциональный блок представляет собой некоторую конкретную функцию моделируемой системы. При этом каждая из четырех сторон функционального блока имеет своё определенное значение (роль) – интерфейсные дуги:

1. входная интерфейсная дуга «Вход» (Input) характеризует материалы или информацию, которые преобразуются функциональным блоком для создания выхода (продукта действия);

2. интерфейсная дуга «Управление» (Control) управляет преобразованием входа в выход;

3. интерфейсная дуга «Механизм» обозначает ресурсы, которые требуются для преобразования входа в выход;

4. выходящая интерфейсная дуга «Выход» (Output) – это материалы или информация, производимая функциональным блоком из входа [5].

На рисунке 2 представлена контекстная диаграмма процесса аттестации медицинских работников с интерфейсными дугами, представленными в таблице 2.

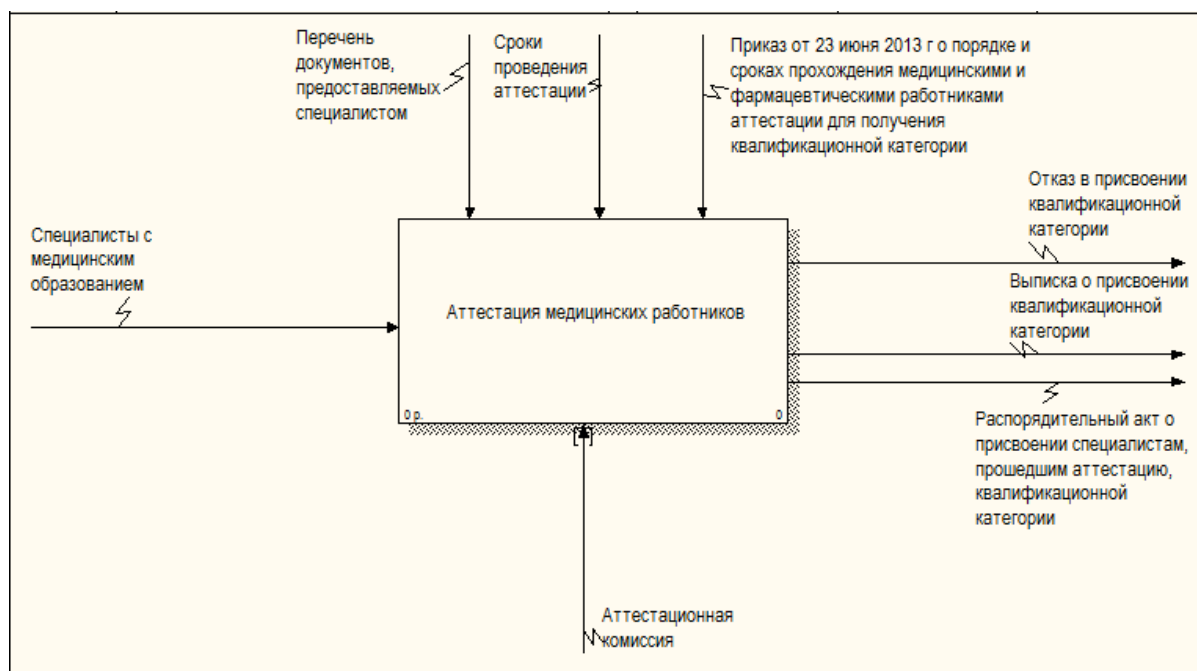


Рисунок 2 – Контекстная диаграмма в методологии IDEF0

В результате декомпозиции функционального блока контекстной диаграммы были определены основные процессы прохождения аттестации медицинских работников (рисунок 3).

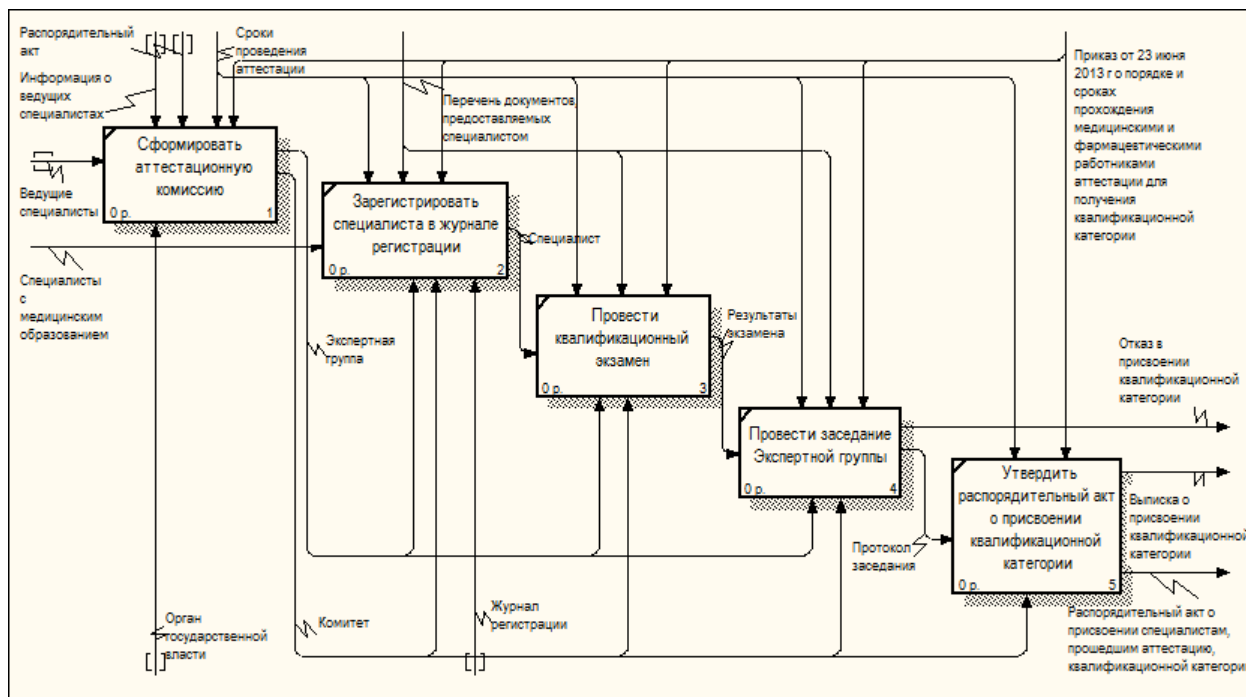


Рисунок 3 – Декомпозиция контекстной диаграммы в методологии IDEF0

Формируется аттестационная комиссия (функциональный блок А1), которая состоит из Координационного комитета (далее – Комитет), осуществляющего функции по организации деятельности аттестационной комиссии, и экспертных групп по специальностям (далее – Экспертные группы), осуществляющих аттестацию специалистов в части рассмотрения документов и проведения квалификационного экзамена.

Секретарь комитета регистрирует и рассматривает поступающие в аттестационную комиссию документы специалистов, изъявивших желание пройти аттестацию для получения квалификационной категории, на предмет соответствия установленным настоящим Порядком и сроками требованиям к перечню и оформлению документов (функциональный блок А2) [2].

После регистрации документов специалистов проводится квалификационный экзамен (функциональный блок А3), по которому

Экспертная группа принимает решение о присвоении квалификационной категории на заседании (функциональный блок А4).

Протокол заседания отправляется в комитет, где готовится и утверждается распорядительный акт о присвоении квалификационной категории (функциональный блок А5).

Взаимодействие функций и процессов между собой описывается интерфейсными дугами, представляющими некую информацию. В таблице 3 и 4 представлена таблица назначений интерфейсных дуг.

Таблица 3 – Таблица словарей интерфейсных дуг контекстной диаграммы А0

Наименование интерфейсных дуг (стрелок)	Их назначение
Перечень документов, которые специалист должен подать, для регистрации на прохождение аттестации (Интерфейсная дуга «Управление»)	Специалисты, изъявившие желание пройти аттестацию для получения квалификационной категории, представляют в аттестационную комиссию с документами
Сроки проведения аттестации (Интерфейсная дуга «Управление»)	Аттестация проводится в фиксированный период времени
Приказ от 23 июня 2013 года о порядке и сроках прохождения медицинскими и фармацевтическими работниками аттестации для получения квалификационной категории (Интерфейсная дуга «Управление»)	Официальный документ о порядке и сроках прохождения аттестации
Специалисты с медицинским образованием, которые будут проходить аттестацию (Входящая интерфейсная дуга «Вход»)	Специалисты, желающие пройти аттестацию на повышение квалификационной категории
Распорядительный акт о присвоении специалистам, прошедшим аттестацию, квалификационной категории (Выходящая интерфейсная дуга «Выход»)	Официальный документ о присвоении квалификационной категории
Выписка о присвоении квалификационной категории, которая предоставляется специалисту (Выходящая интерфейсная дуга «Выход»)	Выдача на руки (направление почтовой связью) выписки из распорядительного акта органа государственной власти
Отказ в присвоении квалификационной категории (Выходящая интерфейсная дуга «Выход»)	Отказ в виду некоторого ряда причин
Аттестационная комиссия (Интерфейсная дуга «Механизм»)	Комиссия, которая организует, проводит квалификационный экзамен и принимает решение о присвоении квалификационной категории

Таблица 4 – Таблица словарей интерфейсных дуг декомпозиции контекстной диаграммы

Наименование интерфейсных дуг (стрелок)	Их назначение
Информация о специалистах (Интерфейсная дуга «Управление»)	Информация о ведущих специалистах
Распорядительный акт (Интерфейсная дуга «Управление»)	Персональный состав аттестационной комиссии утверждается распорядительным актом органа государственной власти или организации, создавших аттестационную комиссию
Повторная регистрация (Выходящая интерфейсная дуга «Выход»)	По результатам устранения оснований, послуживших причиной отказа в принятии документов, специалист вправе повторно направить документы в аттестационную комиссию
Ведущие специалисты (Входящая интерфейсная дуга «Вход»)	Специалисты организаций, осуществляющих медицинскую и фармацевтическую деятельность, представители медицинских профессиональных некоммерческих организаций, работодателей и иные лица
Результат экзамена (Выходящая интерфейсная дуга «Выход»)	Результаты тестирования и собеседования, которые проходил специалист
Экспертная группа (Интерфейсная дуга «Механизм»)	Осуществляет функции по организации деятельности аттестационной комиссии
Комитет (Интерфейсная дуга «Механизм»)	Осуществляет аттестацию специалистов в части рассмотрения документов и проведения квалификационного экзамена
Протокол заседания (Выходящая интерфейсная дуга «Выход»)	Решение Экспертной группы о присвоении или об отказе в присвоении специалисту квалификационной категории оформляется протоколом заседания Экспертной группы
Журнал регистрации (Интерфейсная дуга «Механизм»)	Журнал, куда регистрируют всех специалистов, подавших заявку на присвоение квалификационной категории
Орган государственной власти (Интерфейсная дуга «Механизм»)	Принимают решение о формирование аттестационной комиссии

В результате декомпозиции функционального блока А3 были определены основные процессы прохождения квалификационного экзамена в методологии IDEF3. IDEF3 – методология документирования процессов, происходящих в системе, которая используется, например, при исследовании технологических процессов на предприятиях. С помощью IDEF3 описываются сценарий и последовательность операций для каждого

процесса [5].

Функция «Провести квалификационный экзамен» представлена в виде отдельного процесса средствами IDEF3, представленного на рисунке 4. После регистрации документов Экспертной группой проводится их рассмотрение, утверждается заключение на отчет и назначается дата и место проведения тестового контроля знаний и собеседования.

Тестовый контроль знаний предусматривает выполнение специалистом тестовых заданий и признается пройденным при условии успешного выполнения не менее 70% общего объема тестовых заданий. Если выполнено меньше 70% общего объема тестовых заданий, то специалист пишет заявление в Минздрав Хакасии для предоставления второй попытки тестирования, которая дается единожды.

Собеседование проводится членами Экспертной группы по теоретическим и практическим вопросам профессиональной деятельности специалиста при условии успешного прохождения им тестового контроля знаний.

Результаты квалификационного экзамена формируются в один отчет для заседания Экспертной группы.

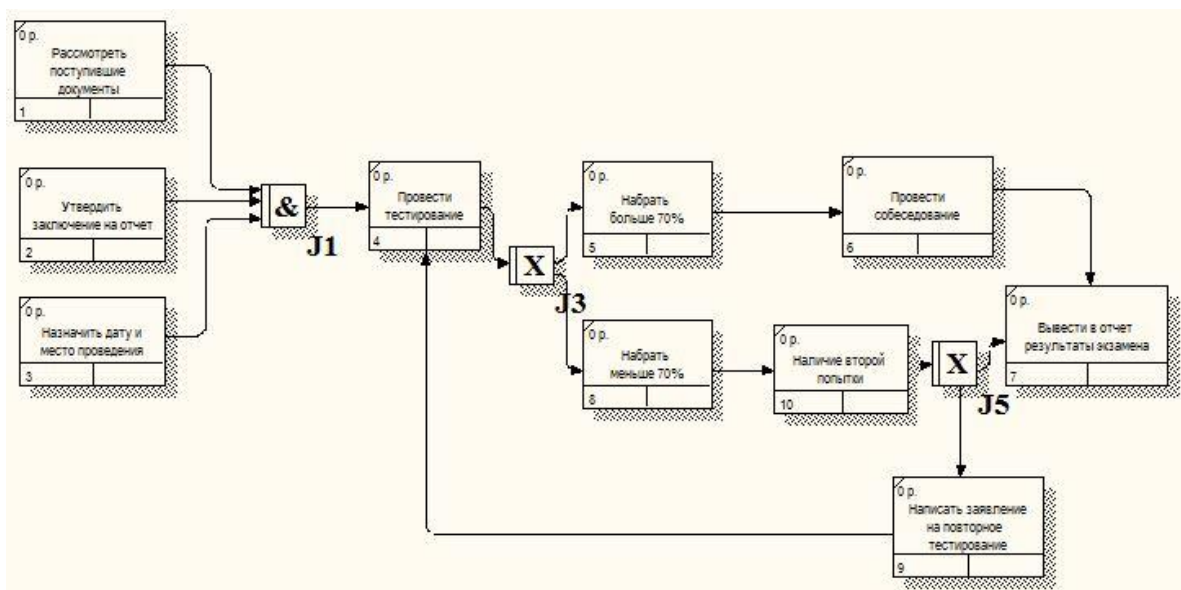





Рисунок 4 – Диаграмма третьего уровня декомпозиции функционального блока А3 в методологии IDEF3

В таблице 5 обозначены типы перекрестов, которые использовались в описании последовательности операций для процесса прохождения квалификационного экзамена.

Таблица 5 – Типы перекрестков диаграммы процесса регистрации клиента

Обозначение	Наименование перекрестка	Описание процесса
	Асинхронное «И»	Запуск все процессов: рассматриваются документы аттестуемого, утверждается отчет по списку аттестуемых, назначается дата и место проведения квалификационного экзамена.
	Исключающий ИЛИ	Возможны два несовместных варианта событий: если тест был завершен с результатом ниже 70%, то аттестуемый не допускается до следующего этапа и, соответственно, не сдает квалификационный экзамен; если тест был завершен с результатом 70% и выше, то аттестуемый допускается до следующего этапа собеседование.
	Исключающий ИЛИ	Возможны два несовместных варианта событий: если у тестируемого есть еще одна попытка на прохождение тестового контроля знаний, то аттестуемый пишет заявление в Минздрав Хакасии; если у тестируемого нет попыток, то аттестуемый завершает квалификационный экзамен.

Таким образом, процесс аттестации медицинских и фармацевтических работников длительный и трудоемкий, так как проводится во всей республике Хакасия. Система аттестационного тестирования медицинских работников сократит время обработки информации и получения результатов тестирования для Аттестационной комиссии.

1.5 Постановка цели и задач проектирования

Министерство здравоохранения Республики Хакасия использует информационную систему аттестационного тестирования медицинских работников «Система тестирования медицинских работников 2.1». Расширение функциональных возможностей не проводилось с 2014 года.

Целью является разработка и тестирование системы аттестационного тестирования медицинских работников республики Хакасия.

Для достижения поставленной цели нужно решить следующие задачи:

1. провести сбор и анализ информации о процессе аттестации медицинских работников Республики Хакасия;
2. построить функциональную модель аттестации медицинских работников;
3. определить цель и задачи проектирования системы аттестационного тестирования;
4. провести анализ существующих систем тестирования и обосновать актуальность разработки информационной системы тестирования медицинских работников своими средствами;
5. провести теоретический анализ и выбрать программные средства разработки информационной системы тестирования медицинских работников;
6. разработать и провести тестирование информационной системы;
7. оценить экономическую эффективность внедрения информационной системы аттестационного тестирования медицинских работников в процесс аттестации.

1.6 Сравнительный анализ существующих систем

Компьютерные системы тестирования знаний классифицируют с использованием семи основных классификационных признаков:

1. по достигаемым в ходе тестирования целям, например, тесты аттестационные, обучающие, информационные и т.д.;
2. по процедуре создания тесты можно подразделять на стандартизованные или не стандартизованные;
3. по способу формирования заданий – детерминированные, стохастические, динамические;
4. по форме предлагаемых тестируемым заданий – закрытого типа, открытого типа;

5. по наличию обратной связи – традиционные системы тестирования или адаптивные системы тестирования;

6. по особенностям технической реализации;

7. по стоимости разработки или приобретения компьютерных систем тестирования [6];

Для системы аттестационного тестирования медицинских работников рассмотрим классификацию по особенностям технической реализации, которые подразделяют компьютерные системы тестирования на две группы: локальные и клиент-серверные. К первой группе будут отнесены программы, работающие на компьютере локально, без необходимости использования возможностей компьютерной сети. Вторая группа – клиент-серверные решения, подразделяющаяся на две подгруппы: в первой для начала тестирования необходимо установить на компьютер тестируемого специальную программу – клиент, взаимодействующую с сервером тестирования. Ко второй подгруппе будут отнесены клиент-серверные решения, в которых тестируемый для ответов использует web-интерфейс обычного веб-браузера [6].

Для разработки системы аттестационного тестирования медицинских работников самым оптимальным решением будет использовать клиент-серверные решения, в которых тестируемый для ответов использует web-интерфейс обычного веб-браузера, что является достаточно хорошим решением, так как не предъявляются дополнительные технические требования к аппаратному и/или программному обеспечению, которые могут возникнуть, если необходимо установить на компьютер специализированную программу-клиент для организации рабочего места тестируемого.

Для сравнения возможностей были отобраны шесть систем тестирования, достаточно часто используемых для организации тестирования знаний: MyTestXPro [8], x-TLS [9], INDIGO [10], Moodle [11], OpenTest [12] и Let's test [13], результаты сравнения которых приведены в таблице 6. Обозначения: «—» значит, что соответствующей возможности в системе

тестирования нет; «+» значит, что соответствующая возможность в системе тестирования имеется.

Таблица 6 – Сравнительный анализ существующих систем тестирования

Возможности	INDIGO	Let's test	MyTestXPro	x-TLS	Moodle	OpenTest
1 Настройка расписания времени проведения тестирования	+	+	—	+	+	+
2 Распространение лицензионной версии	200 соединений (одновременно тестируемых пользователей) – 100 000 рублей	Бесплатная (50 сеансов тестирования), 5 900 рублей/месяц за тестирование без ограничений и дополнительные функции	Пробная версия на 30 дней, цена за копию на 1 компьютер - 400 рублей	Бесплатная с доработкой системы за денежные средства	от 66 200 руб. / в год на оборудование организации	Стоимость зависит от объема тестирования и требуемых функций, которая начинается от 6 тыс. рублей.
3 Клиент-серверное решение	+	+	+	+	+	+
4 Доработка модулей программы, интегрирование собственных блоков	—	—	—	+	—	—
5 Экспорт таблиц с результатами	+	+	+	—	+	+
6 Регистрация пользователей	Регистрирует администратор или пользователь самостоятельно регистрируется через web-интерфейс	Пользователь самостоятельно регистрируется, в платной версии добавление пользователей администратором	Пользователь самостоятельно регистрируется	Пользователь самостоятельно регистрируется	Регистрирует администратор	Регистрирует администратор
7 Установление дополнительного анкетирования	—	+	—	—	+	+
8 Установка требуемого числа попыток для участников тестирования	+	+	—	—	+	+

Анализ представленных данных позволяет сделать вывод:

1. рассмотренные выше системы тестирования INDIGO, MyTestXPro и x-TLS не имеют возможности установки дополнительного анкетирования, что является обязательным требованием для системы аттестационного тестирования, также в системах MyTestXPro и x-TLS нет возможности установки количества попыток;

2. системы тестирования Let's test и OpenTest располагаются на серверах компании производителя систем, что является неприемлемым для разрабатываемой системы аттестационного тестирования медицинских работников;

3. система тестирования Moodle отвечает требованиям, но имеет большой набор функционала, который не будет использоваться специалистами для аттестационного тестирования и увеличивают временные и денежные затраты на дополнительные настройки.

1.7 Обоснование выбора средств разработки информационной системы аттестационного тестирования

Анализ возможностей и функционирования существующих систем и требований организации для создания информационной системы аттестационного тестирования позволяет выбрать технологию создания информационной системы как веб-сервер.

Web-приложения представляют собой особый тип программ, построенных по архитектуре "клиент-сервер". Особенность их заключается в том, что само Web-приложение находится и выполняется на сервере - клиент при этом получает только результаты работы. Работа приложения основывается на получении запросов от пользователя (клиента), их обработке и выдачи результата. Передача запросов и результатов их обработки происходит посредством сети (рисунок 5) [7].

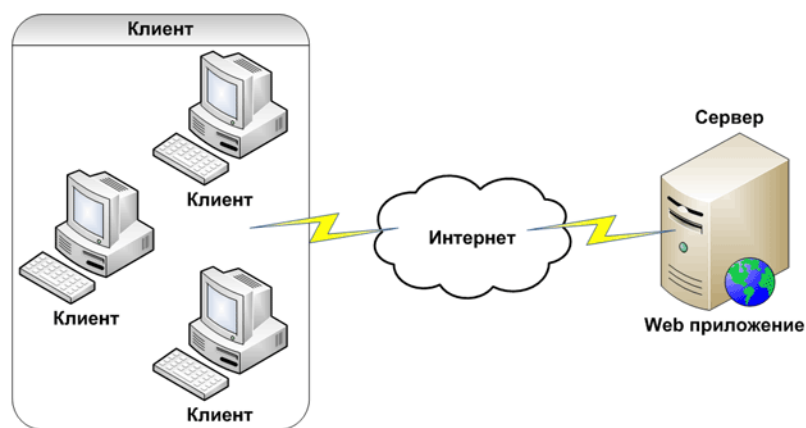


Рисунок 5 – Архитектура web-приложения

Отображением результатов запросов, а также приемом данных от клиента и их передачей на сервер обычно занимается специальное приложение — браузер. Как известно, одной из функций браузера является отображение данных, полученных из Интернета, в виде страницы, описанной на языке HTML, следовательно, результат, передаваемый сервером клиенту, должен быть представлен на этом языке [7].

Для повышения безопасности передачи персональных данных медицинских работников используется расширенный протокол http – https. Секретность работы в Интернете обеспечивается путём организации защищённого шифрования канала связи между клиентским браузером и тем узлом сети (web-сервером), с которым пользователь браузера обменивается информацией.

1.7.1 Выбор информационного обеспечения и его обоснование

Разработка информационной системы по технологии веб-сервис, требует создания централизованной базы данных (БД) с соответствующей архитектурой клиент-сервер. Для создания БД необходимо приложение (или библиотеки), управляющие базами данных различных форм, размеров и типов – СУБД [14].

Сравним несколько СУБД, широко используемых в Интернет: SQLite, SQL Server, MySQL.

SQLite – библиотека, встраиваемая в приложение, которое её использует. Будучи файловой БД, она предоставляет отличный набор инструментов для более простой (в сравнении с серверными БД) обработки любых видов данных.

Когда приложение использует SQLite, их связь производится с помощью функциональных и прямых вызовов файлов, содержащих данные, а не какого-то интерфейса, что повышает скорость и производительность операций [14].

Преимущества SQLite в том, что:

1. вся БД хранится в одном файле, что облегчает перемещение;
2. стандартный функционал SQL.

Главным недостатком именно для разработки системы аттестационного тестирования то, что она не предназначена для многопользовательских приложений. SQLite стоит использовать, когда для разработки встроенных приложений (локальных однопользовательских приложений).

Microsoft SQL Server — система управления реляционными базами данных (СУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов – Transact-SQL. Используется для работы с небольшими и средними по размеру базами данных до крупных баз данных масштаба предприятия [19].

Microsoft SQL Server Express является бесплатно распространяемой версией SQL Server, которая имеет некоторые технические ограничения. Такие ограничения делают её непригодной для развертывания больших баз данных, но она вполне годится для ведения программных комплексов в масштабах небольшой компании. Содержит полноценную поддержку новых типов данных. Фактически, это полноценный MS SQL Server, включая все его компоненты программирования, поддержку национальных алфавитов и Unicode [19].

MySQL – это самая популярная из всех крупных серверных БД. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон [14]. Это система с открытым исходным кодом. Открытость исходного кода означает, что любой желающий имеет возможность использовать и модифицировать это программное обеспечение по своему усмотрению. Получить MySQL можно бесплатно.

MySQL имеет ряд преимуществ:

1. поддерживает большое количество типов данных;
2. существует много сторонних инструментов, включая визуальные, облегчающих начало работы с БД;
2. MySQL поддерживает большую часть функционала SQL;
3. может работать с действительно большими объёмами данных, и походит для масштабируемых приложений;
4. многопользовательские приложения.

Согласно проведенному анализу СУБД проектирование БД будет осуществляться с помощью MySQL. Выбор данной СУБД обусловлен также тем, что благодаря популярности и распространенности MySQL, сокращаются финансовые затраты на поддержку (точнее бесплатно) и временные затраты на поиск специалиста. Также СУБД MySQL является клиент-серверной системой, включающей многопоточный SQL-сервер, поддерживающий различные платформы, несколько клиентских программ и библиотек, инструменты администрирования и широкий диапазон программных интерфейсов приложений [19].

1.7.2 Обоснование выбора языка программирования для разработки информационной системы

Современные веб-приложения являются интерактивными и чаще всего для их разработки современные IT-специалисты используют языки

программирования: PHP, Python и Ruby. Охарактеризуем каждый из них и обоснуем выбор языка программирования для разработки информационной системы тестирования медицинских работников.

Язык программирования *Ruby* – это динамический императивный объектно-ориентированный язык программирования с открытым исходным. Используется в веб-разработке в составе открытого веб-фреймворка Rails, чаще называемого Ruby on Rails (RoR) [16].

Ruby – это тщательно сбалансированный язык, который объединил части языков (Perl, Smalltalk, Eiffel, Ada и Lisp) [17] и имеет цель приблизить синтаксис реальной программы, написанной на нём, к описывающему задачу псевдокоду, что позволяет программисту уменьшить объём программы [16].

Файлы программ, написанных на Ruby, имеют расширение .rb. Эти файлы можно запустить как скрипты командной оболочки при помощи интерпретатора [16].

В Ruby всё – объект. Для каждой частицы информации или кода могут быть определены собственные свойства и действия [17].

Главной особенностью языка программирования Ruby является то, что это конструктор, из которого собирается программный код.

Преимущества языка программирования Ruby:

1. открыт для бесплатного использования, изменений, копирования и распространения;
2. возможность внедряться в HTML-разметку;
3. возможности языка можно расширить при помощи библиотек, написанных на C или Ruby;
4. легко интегрирует в свои программы высокопроизводительные серверы баз данных (DB2, MySQL, Oracle и Sybase);
5. отличается высокой переносимостью, но он был разработан большей частью на GNU/Linux, но работает на многих типах UNIX, Mac OS X, Windows, DOS, BeOS, OS/2, и так далее;
6. не надо объявлять переменные.

Язык программирования Ruby менее производителен по сравнению со многими другими языками, применяемыми в веб-разработке, что является недостатком. Также вследствие недостатков: неуправляемость некоторых процессов (таких, как выделение памяти), невозможность задания низкоуровневых структур данных или подпрограмм и невозможность компиляции и сопутствующей ей оптимизации программы – программа, написанная на языке Ruby, имеет весьма низкую скорость запуска и выполнения.

Еще один язык программирования, целью которого является уменьшить объем программы, – *Python*. Это мощный инструмент для создания программ самого разнообразного назначения [18].

Python – интерпретируемый язык программирования. С одной стороны, это позволяет значительно упростить отладку программ, с другой – обуславливает сравнительно низкую скорость выполнения [16].

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных [16].

Преимущества языка программирования Python:

1. множество полезных библиотек и расширений языка;
2. предельно унифицированный механизм импорта;
3. огромное количество модулей;
4. программа, написанная на Python, будет функционировать совершенно одинаково вне зависимости от того, в какой операционной системе она запущена.

Недостатки языка программирования Python: ограниченность средств для работы с базами данных, создано не так уж много качественных

программных проектов по сравнению с другими универсальными языками программирования.

Язык программирования *PHP* достаточно гибкий, поэтому имеет большую популярность и используется в проектах любого масштаба: от простого блога до крупнейших веб-приложений в Интернете.

PHP (рекурсивный акроним словосочетания PHP: Hypertext Preprocessor) – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP сконструирован специально для ведения Web-разработок и его код может внедряться непосредственно в HTML [15].

Скрипты, написанные на языке PHP, обычно хранятся в файлах с расширением .php, которые содержат в себе смесь обычных HTML-тэгов со специальной разметкой: открывающим тэгом `<?php` и закрывающим `?>` [15].

Существуют три основных области применения PHP:

1. создание скриптов для выполнения на стороне сервера (для просмотра результатов выполнения PHP-скриптов в браузере, нужен работающий веб-сервер и установленный PHP);
2. создание скриптов для выполнения в командной строке (PHP-скрипт, способный запускаться без сервера или браузера);
3. создание оконных приложений, выполняющихся на стороне клиента [15].

Но главной областью применения PHP является написание скриптов, работающих на стороне сервера, таким образом PHP способен обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies (небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя).

Преимуществом языка программирования PHP является:

1. доступность для большинства операционных систем, включая Linux, Unix, Microsoft Windows, Mac OS X, RISC OS и многие другие;

2. поддержка большинства современных веб-серверов, таких как Apache, IIS и многих других;
3. осуществляет автоматическую генерацию файлов PDF, любых текстовых данных, изображений и сохранять их в файловой системе сервера;
4. поддерживает широкий круг баз данных (MySQL, PostgreSQL, Sybase, Informix и др.);
5. имеется огромное количество библиотек и расширений языка;
6. является свободным программным обеспечением [15].

Таким образом, при выборе языка PHP, появляется возможность свободы выбора операционной системы, веб-сервера и базы данных. Также язык PHP прост в освоении, удобен в работе с запросами SQL, имеет высокую скорость выполнения и богатую функциональность.

Кроме того, для разработчиков на языке PHP существует большое количество литературы, как в печатном, так и в электронном варианте, веб-сайтов, различных форумов и блогов.

Согласно проведенному анализу самым оптимальным и удобным языком программирования для разработки системы аттестационного тестирования является PHP, так как является свободным программным обеспечением, поддерживает современные СУБД, веб-сервера и операционные системы и непосредственно PHP интегрирован для разработки систем по типу архитектуры «клиент-сервер», упрощает такие задачи, как поиск хостинга, специалиста по поддержке и уменьшает затраты на сопровождение.

В разработке будет использоваться язык разметки гипертекста HTML, так как он является одним из основных средств создания веб-страниц, для декорации страницы – CSS и JavaScript – язык, отвечающий за поведение веб-страницы.

HTML – популярный язык гипертекстовой разметки документов, позволяющий создавать интерактивные публикации в Интернете [20].

Cascading Style Sheets (Таблицы Каскадных Стилей) - это язык, содержащий набор свойств для описания внешнего вида любых HTML документов. С его помощью дизайнер имеет полный контроль над стилем и расположением каждого элемента веб страницы, что проще и гораздо функциональнее использования обычного набора HTML тегов [20].

С помощью JavaScript можно легко создавать интерактивные Web-страницы [20].

Языкам JavaScript, HTML и CSS не существует альтернативы.

Таким образом, для разработки системы аттестационного тестирования были выбраны СУБД MySQL, язык программирования PHP, язык разметки гипертекста HTML, CSS и JavaScript.

Вывод по аналитическому разделу

В аналитическом разделе ВКР основные задачи Министерства здравоохранения Республики Хакасия, как полномочного присваивать квалификационные категории медицинским работникам.

Одной из задач Министерства здравоохранения Республики Хакасия, решение которой направлено на повышение качества медицинского обслуживания населения является проведение аттестации медицинских и фармацевтических работников, рассмотренной более подробно в методологиях IDEF0 и IDEF3.

В рамках ВКР проводится анализ процесса тестирования медицинских работников в соответствии с современными требованиями и техническими средствами реализации подобных систем. В настоящее время существующая система «Система тестирования медицинских работников 2.1» не соответствует требованиям, которые к системе предъявляются на современном этапе, появились новые задачи аттестационного тестирования и потребность в улучшении функционала.

Рассмотрена возможность приобретения готового продукта – системы тестирования. Однако, последующий анализ рынка существующих популярных систем тестирования INDIGO, OpenTest, let's test, MyTestXPro, x-TLS, Moodle позволил сделать вывод: для аттестации медицинских работников данные системы либо не подходят, так как не имеют требуемый функционал, либо имеют довольно обширный перечень возможностей, который не требуется для системы аттестационного тестирования медицинских работников (система Moodle).

Принятое решение разработать новую информационную систему аттестационного тестирования медицинских работников потребовало анализа средств разработки и обоснования их выбора. Для реализации информационной системы аттестационного тестирования медицинских работников были проанализированы наиболее популярные средства в среде современных IT-разработчиков

1. языки программирования: PHP, Ruby, Python;
2. СУБД MySQL, Microsoft SQL Server, SQLite.

Для проектирования БД выбрана СУБД MySQL, так как она не требует денежных затрат и является клиент-серверной системой.

Для разработки системы аттестационного тестирования медицинских работников выбран язык программирования PHP, который имеет возможность свободы выбора операционной системы, веб-сервера и базы данных, удобен в работе с запросами SQL, имеет высокую скорость выполнения и богатую функциональность.

Также в разработке будет использоваться язык разметки гипертекста HTML, для декорации страницы – CSS и JavaScript – язык, обеспечивающий поведение веб-страниц.

2 Практическая реализация разработки информационной системы аттестационного тестирования медицинских работников (модули администратора и модератора)

В практическом разделе ВКР реализована задача проектирования и разработки модулей администратора и модератора информационной системы аттестационного тестирования медицинских работников.

2.1 Проектирование модуля администратора и модератора системы аттестационного тестирования медицинских работников

В пункте 2.1 практического раздела смоделированы потоки данных, на основании которых спроектирована структура базы данных.

2.1.1 Определение и анализ данных, необходимых для выполнения каждой функции процесса

Для отображения как каждый процесс модуля администратора преобразует свои выходные данные в выходные, а также для выявления отношений между этими процессами представим их в методологии DFD (Data Flow Diagrams) – диаграммы потоков данных, которые представляют собой иерархию функциональных процессов, связанных потоками данных.

Основными компонентами DFD являются

1. внешние сущности (материальный объект или физическое лицо, являющиеся источником или приемником информации);
2. процессы – преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом;
3. накопители данных – абстрактное устройство для хранения информации;

4. потоки данных определяет информацию, передаваемую через некоторое соединение от источника к приемнику [22].

Контекстная диаграмма модуля администратора системы аттестационного тестирования медицинских работников построена на рисунке 6, на которой отображены взаимосвязи внешних сущностей и модуля администратора. Внешние сущности представлены: администратор и Минздрав Хакасии.

Администратор в системе аттестационного тестирования медицинских работников с данными пользователя и его результатами, изменяет и дополняет информацию, а Минздрав Хакасии предоставляет список медицинских работников, которые будут проходить тестирование в системе, и получает результаты их тестирования.

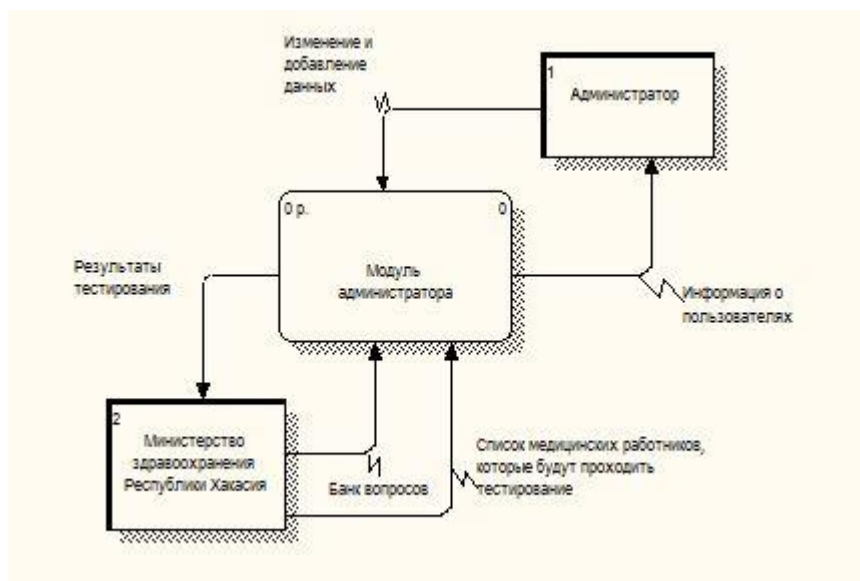


Рисунок 6 – Контекстная диаграмма модуля администратора в методологии DFD

Взаимосвязанная работа внешних сущностей с модулем представлена потоками данных (стрелками на диаграмме), описанных в таблице 7.

Таблица 7 – Потоки данных процесса функций администратора (контекстная диаграмма)

Наименование потоков данных	Описание
Изменение и добавление данных	Возможность изменять, добавлять данных, которые используются в системе
Информация о пользователях	Вывод данных о пользователях
Результаты тестирования	Результат тестирования в процентах
Банк вопросов	Перечень вопросов, из которых составляется тест
Список медицинских работников, которые будут проходить тестирование	Список ожидаемых тестируемых

Для отображения полного функционала администратора в соответствующем модуле сделана декомпозиция контекстной диаграммы, которая представлена на рисунке 7, на котором подробно отражено взаимодействие между процессами и накопителями данных, входные и выходные данные.

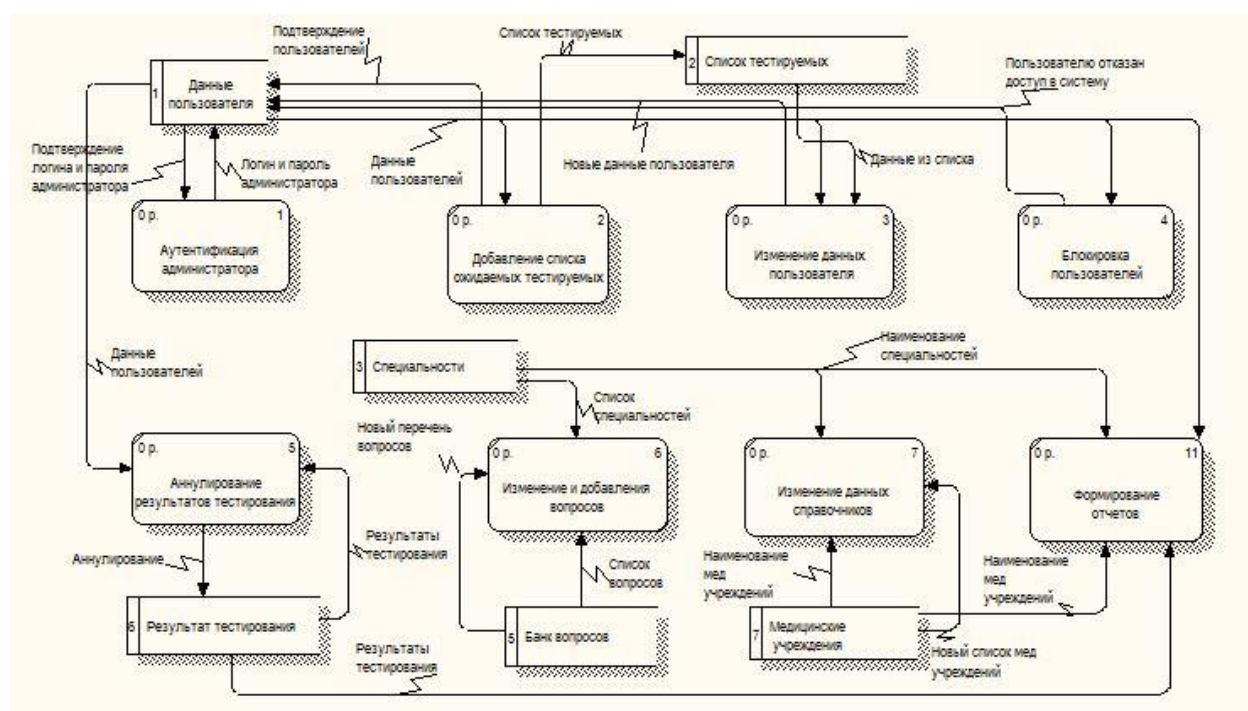


Рисунок 7 – Декомпозиция контекстной диаграммы модуля администратора в методологии DFD

В декомпозиции контекстной диаграммы модуля администратора выделены следующие процессы:

1. аутентификация администратора;
2. добавление списка ожидаемых тестируемых;
3. изменение данных пользователя;
4. блокировка пользователей;
5. аннулирование результатов тестирования, изменение и добавление вопросов;
6. изменение данных справочника;
7. формирование отчетов.

В диаграмме представлены следующие накопители данных: данные пользователя, список ожидаемых тестируемых, банк вопросов, специальности, результаты тестирования, медицинское учреждения.

Взаимосвязи между процессами и накопителями данных представлены потоками данных (стрелки), которые отражают входные и выходные данные. Подробно описаны потоки данных в таблице 8.

Таблица 8 – Потоки данных процесса функций администратора (декомпозиция контекстной диаграммы)

Наименование потоков данных	Описание
Подтверждение логина и пароля администратора	Подтверждение введенного логина и пароля с логином и паролем администратора, хранящейся в накопителе
Логин и пароль администратора	Введенные логин и пароль для входа в модуль администратора
Подтверждение пользователей	При регистрации пользователя происходит подтверждение того, что данный пользователь есть в списке ожидаемых
Данные пользователей	Данные пользователей введенные при регистрации
Список тестируемых	Список медицинских работников, которые будут проходить тестирование
Новые данные пользователя	Измененные данные пользователя обновляются в накопителе
Данные из списка	Конкретные данные, по которым идет подтверждение пользователя

Окончание таблицы 8

Пользователю отказан доступ в систему	Запрет на вход в систему пользователю
Аннулирование	Предоставление пользователю еще одной попытки для прохождения тестирования с сохранением предыдущего результата
Список вопросов	Список вопросов, из которых формируется тест
Новый перечень вопросов	Добавленные или обновленные вопросы
Список специальностей	Специальности в области медицины, по которым проводится тестирование
Наименование специальностей	Перечень официальных наименований специальностей

Для отображения процессов модуля модератора (сотрудника Министерства здравоохранения республики Хакасия) используется аналогичная диаграмма потоков данных, что и для администратора, но без процессов:

1. изменение данных пользователя;
2. блокировка пользователей;
3. изменение данных справочников.

Контекстная диаграмма модуля модератора представлена на рисунке 8. Внешней сущностью является модератор, который непосредственно вводит банка вопросов для тестирования и список ожидаемых тестируемых, просматривает результаты тестирования.

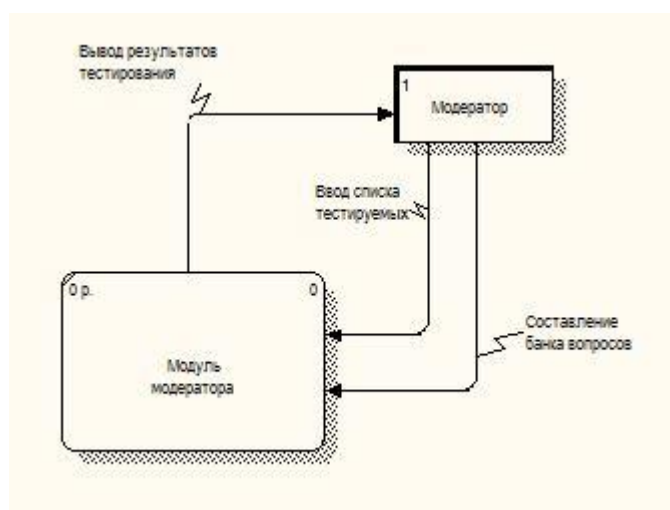


Рисунок 8 – Контекстная диаграмма модуля модератора в методологии DFD

Декомпозиция контекстной диаграммы модуля модератора представлена на рисунке 9.

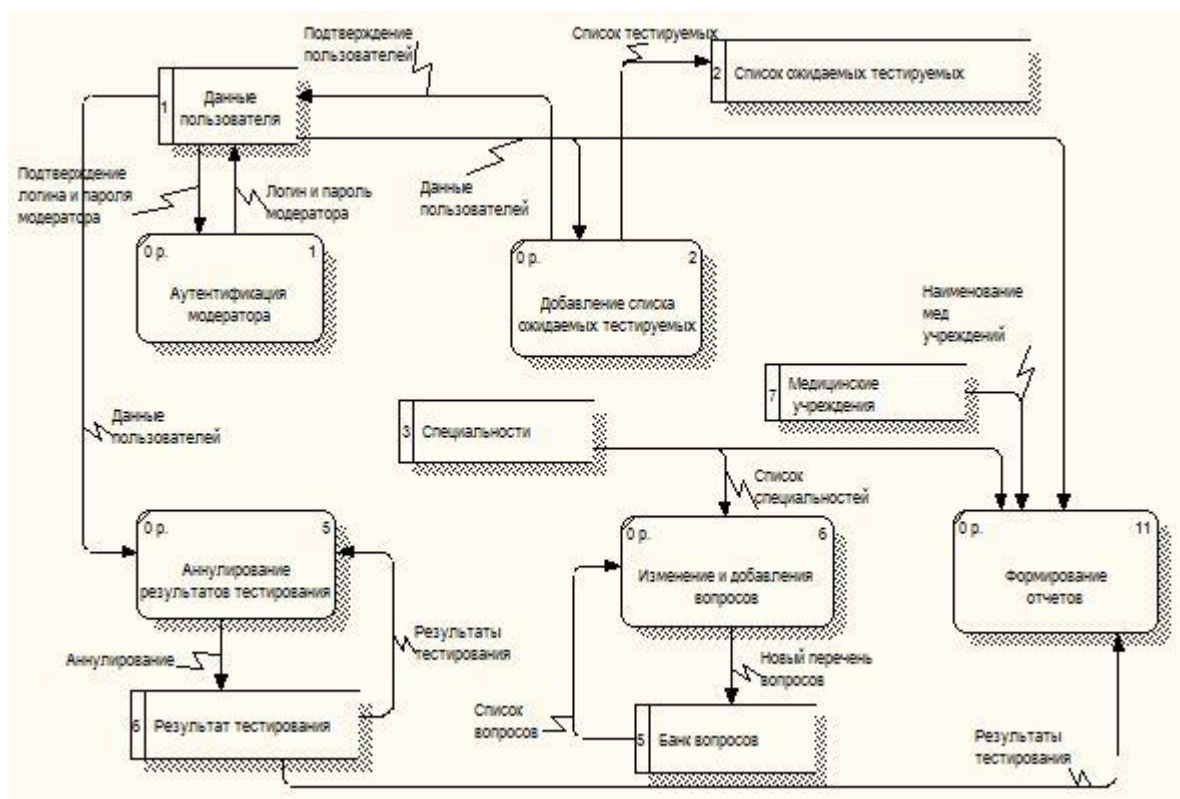


Рисунок 9 – Декомпозиция контекстной диаграммы модуля модератора в методологии DFD

Таким образом, процессы модуля модератора позволяют сократить время ввода и обработки информации, так как нет необходимости связываться напрямую со специалистом ГКУЗ РХ «РМИАЦ», отвечающим за систему тестирования для получения доступа к результатам тестирования.

2.1.2 Проектирование модели структуры данных

В результате построения диаграмм потоков данных, представленных на рисунках 6, 7, 8 и 9, выявлены накопители, которые будут составлять логическую схему базы данных системы аттестационного тестирования.

При создании логической модели БД является необходимым построение диаграммы ERD (Entity Relationship Diagram). ERD-диаграммы состоят из трех частей: сущностей, атрибутов и взаимосвязей.

В БД будут храниться записи о медицинских работниках, следовательно, сущностью будет медицинский работник, представленная в таблице 9.

Таблица 9 – Атрибуты сущности «Медицинские работники»

Атрибут	Описание
Номер	Уникальный номер для идентификации пользователя
Тип пользователя	Разграничение прав доступа
Логин	Логин для доступа в систему
Пароль	Пароль для доступа в систему
Электронная почта	Адрес электронной почты
Фамилия	Фамилия пользователя
Имя	Имя пользователя
Отчество	Отчество пользователя
Группа специальностей	Список групп специальностей, по которым разделяют весь медицинский персонал
Специальность	Специальность, по которой осуществляет свою деятельность пользователь
Категория	Категория квалификации пользователя
Стаж работы	Продолжительность работы по трудовому договору и по определенной специальности
Место работы	Медицинское учреждение, где работает пользователь
Телефон	Номер телефона
Пробное тестирование	Наличие прохождения пробного тестирования
Основное тестирование	Наличие прохождения основного тестирования
hash	Формирование хеш для подтверждения электронной почты пользователя
confirm	Подтверждение, что в списке ожидаемых тестируемых есть данный пользователь
session	Сессия пользователей

В полученном списке существуют атрибуты, которые являются справочниками, поэтому выделяются в отдельные сущности. К таким атрибутам относятся атрибуты «Группа специальностей», «Специальность»,

«Категория», «Место работы», которые соответственно представлены в таблицах 10, 11, 12 и 13.

Таблица 10 – Атрибуты сущности «Группа специальностей»

Атрибут	Описание
Номер группы специальностей	Уникальный номер для идентификации группы
Группа	Наименование группы

Таблица 11 – Атрибуты сущности «Специальности»

Атрибут	Описание
Номер специальности	Уникальный номер для идентификации специальности
Номер группы специальности	Список специальностей зависит от группы специальностей
Специальность	Наименование специальности

Таблица 12 – Атрибуты сущности «Категория»

Атрибут	Описание
Номер категории	Уникальный номер для идентификации категории
Категория	Наименование категории

Таблица 13 – Атрибуты сущности «Место работы»

Атрибут	Описание
Номер места работы	Уникальный номер для идентификации места работы
Медицинское учреждение	Наименование медицинского учреждения

Так как зарегистрированный пользователь буде проходить тестирование, необходимо хранить данные о тестировании. Выделены следующие атрибуты сущности «Тест»: вопрос, ответ пользователя, тип теста. В таблице 14 представлена сущность «Тест» с атрибутами и их назначением.

Таблица 14 – Атрибуты сущности «Тест»

Атрибут	Описание
Номер	Уникальный номер для идентификации теста
Вопрос	Вопросы тестирования
Ответ пользователя	Выбранный ответ пользователем
Тип теста	Пробный или основной тест

В полученном списке существуют атрибуты, которые нельзя определить в виде одного поля БД. Атрибут «Вопрос» рассматривается как сущность со своими атрибутами, которые представлены в таблице 15.

Таблица 15 – Атрибуты сущности «Вопросы»

Атрибут	Сущность
Номер	Уникальный номер для идентификации вопроса
Вопрос	Вопрос тестов
Тип вопроса	Одиночный ответ или ответ с множественным выбором
Специальность	Специальность, к которой относится вопрос

Банк вопросов зависит от выбранной специальности, но также возможно, что один вопрос может быть для нескольких специальностей, поэтому атрибут «Специальность» выделяется в отдельную сущность «Вопросы по специальности», представленную в таблице 16.

Таблица 16 – Атрибуты сущности «Вопросы по специальности»

Атрибут	Сущность
Номер	Уникальный номер для идентификации
Вопрос	Вопрос
Специальность	Специальность, к которой относится вопрос

Сущность «Ответы» имеет взаимосвязь со сущностью «Вопросы», так как на один вопрос дается несколько вариантов ответов и указывается правильный ответ, так же возможен множественный ответ. В таблице 17 представлена сущность «Ответы» с атрибутами.

Таблица 17 – Атрибуты сущности «Ответы»

Атрибут	Сущность
Номер	Уникальный номер для идентификации ответа
Ответ	Ответ по определенному вопросу
Номер вопроса	Номер вопроса для идентификации соответствующих ответов
Правильный ответ	Верный ответ вопроса

В таблице 18 представлены сущность «Результаты теста», которая взаимосвязана с сущностью «Тест» через атрибут «Номер теста».

Таблица 18 – Атрибуты сущности «Результаты теста»

Атрибут	Сущность
Номер теста	Номер теста для идентификации соответствующих результата
Дата	Дата тестирования
Пользователь	Пользователь, который прошел тест
Тест	Тест, состоящий из 75 вопросов
Результат	Полученный результат тестирования
Процент	Результат в процентах

В таблице 19 представлена сущность «Список ожидаемых тестируемых», которая не связана с другими сущностями, поэтому является свободной сущностью.

Таблица 19 – Атрибуты сущности «Список ожидаемых тестируемых»

Атрибут	Сущность
Номер	Номер для идентификации соответствующего медицинского работника, которого ожидают на тестирование
Фамилия	Фамилия
Имя	Имя
Отчество	Отчество
Место работы	Медицинское учреждение
Дата	Дата внесения данных

На рисунке 10 представлена логическая схема данных, где все ключевые атрибуты будут находиться над горизонтальной чертой внутри рамки, изображающей сущность.

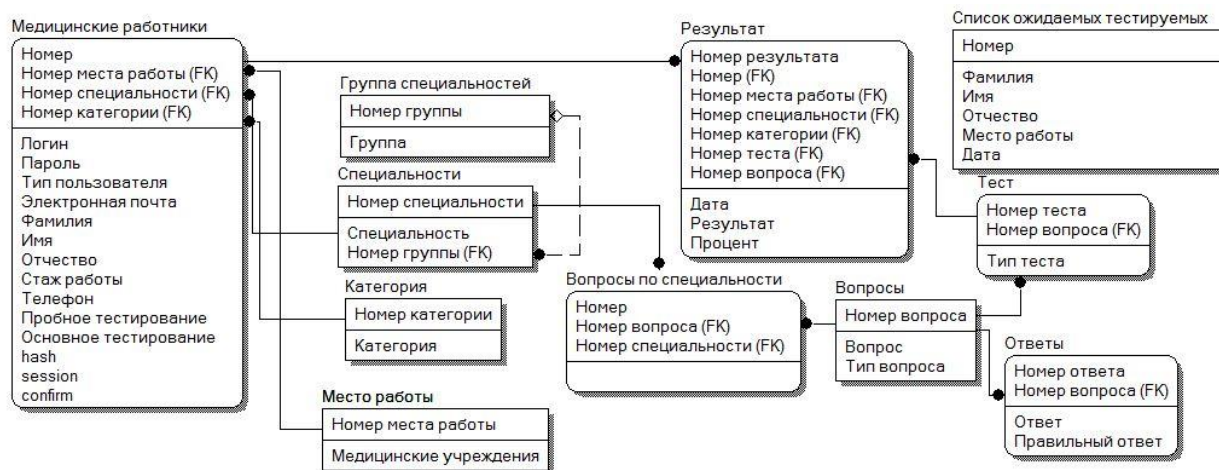


Рисунок 10 – Логическая модель данных, построенная на основе методологии ERD

На логической модели данных связь между сущностями имеет два вида: идентификационная связь один-ко-многим (сплошная линия) и не идентифицирующая связь один-ко-многим (пунктирная линия).

Идентифицирующая связь устанавливается между независимой (родительский) и зависимой (дочерний) сущностями. При установлении неидентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности переносятся в состав неключевых атрибутов родительской сущности, которые в дочерней сущности помечаются как внешний ключ (FK).

Для переноса логической модели данных в СУБД создается физическая модели данных, при этом логическая модель трансформируется в физическую по следующему принципу: сущности становятся таблицами, атрибуты становятся столбцами, а ключи становятся индексами.

На рисунке 11 представлена модель, в которой скорректированы типы и размеры полей, введены правила валидации колонок, определяющие списки допустимых значений и значения по умолчанию.

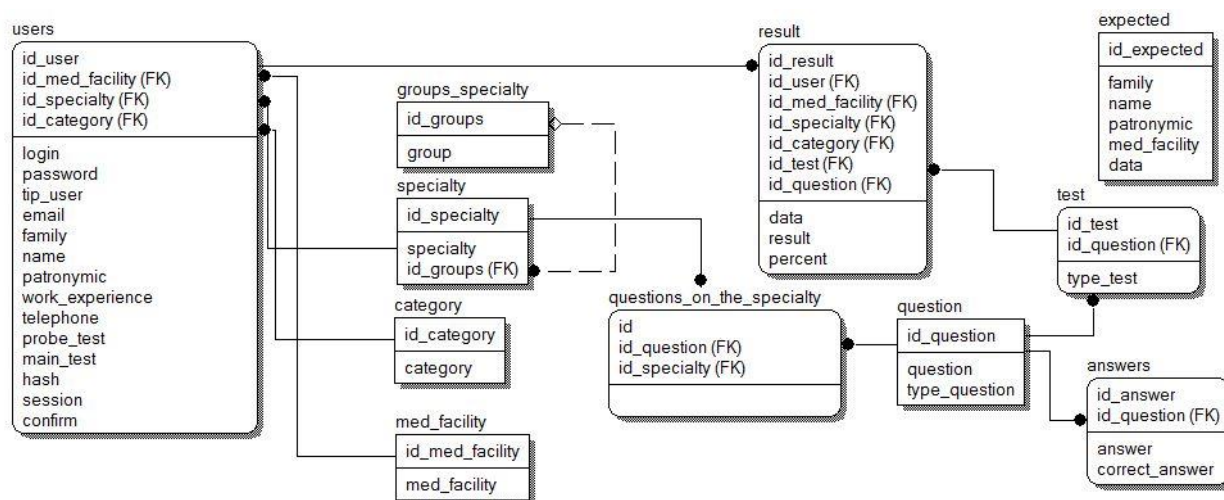


Рисунок 11 – Физическая модель данных, построенная на основе методологии ERD

Таким образом, проделав все вышеописанные действия, полученная модель БД создается в выбранной СУБД MySQL.

2.2 Программно-техническое обеспечение разработки модуля администратора и модератора

В пункте 2.2 практического раздела описана разработка модулей администратора и модератора информационной системы аттестационного тестирования медицинских работников.

2.2.1 Файловая структура модуля администратора и модератора

Для управления системой аттестационного тестирования медицинских работников используется 2 модуля:

1. администратор – модуль, предназначенный для управления системой аттестационного тестирования: добавление и изменение данных пользователей, блокирование пользователей, обнуление результатов тестирования, вывод общей информации по пользователям и результатам тестирования, добавление и изменение банка вопросов;

2. модератор – модуль, в котором реализованы следующие функции: вывод общей информации по пользователям и результатам тестирования, добавление и изменение банка вопросов, изменение и добавление данных пользователей.

Реализация модулей представляет собой набор скриптов и веб-страниц с интегрированными блоками функционального кода PHP, также вспомогательный элемент системы (таблицы каскадных стилей) прописан в «style.css», который отвечает за внешний вид системы.

Процедура подключения к базе данных прописана в файле «config.php», код которого находится в приложении А.

Модули администратора и модератора содержат веб-страницы и скрипты для управления системой, просмотра информации в таблицах базы данных. Главной страницей модулей является страница «../adminresult» (adminresult.php), код которого находится в приложении Б, содержащая HTML-формы и ссылки на другие страницы модуля, а главным скриптом – «functions.php», где содержатся функции обработки HTML-форм для управления данными в БД. В приложении У представлен код обработки данных файла «functions.php».

Модуль модератора аналогичен модулю администратора. И так как модули имеют одинаковую структуру веб-страниц за исключением функций, отображенных в диаграмме потоков данных,

Но для того, чтобы при входе в систему отображались именно те веб-страницы, которые необходимы для каждого модуля соответственно, происходит следующая проверка:

```

if (isset($_POST['login']) && isset($_POST['password'])) {
    $msg = login($_POST);
    if ($msg === TRUE) {
        if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
            header("Location: adminresult.php");
        }
        else { header("Location: kabinet.php");}
    }
    else { $_SESSION['msg'] = $msg;
        header("Location:".$_SERVER['PHP_SELF']);}
    exit(); }

```

Получается в начале файла «header.php», код которого представлен в приложении Т, идёт проверка на тип пользователя через введенный логин. Если это не администратор и не модератор, то есть сессионная переменная «а» не равно 1 или 2 соответственно, то определяется как пользователь системы и переправляется в кабинет пользователя, соответственно у него нет доступа к модулям администратора и модератора.

Если же сессионная переменная «а», равная 1 или 2, подтверждается, то загружается соответствующее модулю администратора и модератора соответственно главная страница – результаты тестирования пользователей – и главное меню.

Данная проверка проходит в начале каждого файла модуля.

2.2.2 Структура страниц модуля администратора и модератора

Разметка главной страницы системы тестирования поделена на 3 части: верхняя часть (шапка), к которой расположено меню, основная часть (контент) и нижняя часть (подвал), за каждой из которых отвечает один php-файл:

1. «header.php» – верхняя часть (шапка) системы, которая в зависимости от действий пользователя меняет внешний вид, представленная на рисунке 12;

2. «content.php» – основная часть, в которой отображаются основные действия пользователя и выводится информация из БД, представленная на рисунке 13;

3. «footer.php» – нижняя часть (подвал), в которой содержится телефон технической поддержки и наименование ГКУЗ РХ «РМИАЦ», представленная на рисунке 14;

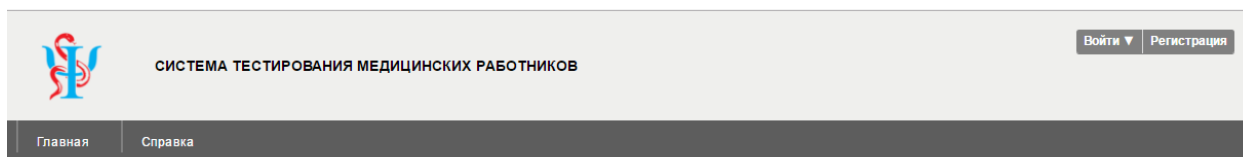


Рисунок 12 – Верхняя часть (шапка) системы

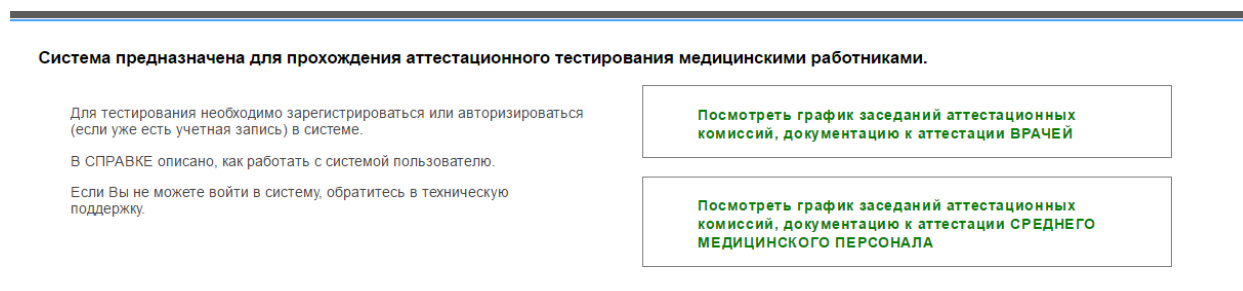


Рисунок 13 – Основная часть системы

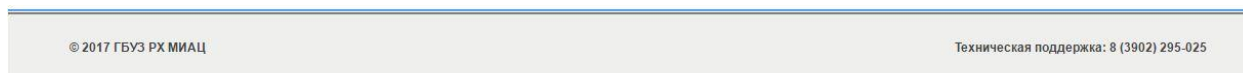


Рисунок 14 – Нижняя часть (подвал) системы

Для входа в систему в верхней части (шапка) сайта справа расположено всплывающее меню с двумя пунктами «Войти» и «Регистрация» (рисунок 12).

Для администратора и модератора доступны следующие страницы:

1. .../adminresult (adminresult.php) – страница вывода результатов всех пользователей в табличном виде, код представлен в приложении Б;

2. .../adminusers (adminusers.php) – страница вывода информации о всех пользователях, код представлен в приложении В;
3. .../ adminchangeuser (adminchangeuser.php) – страница изменения данных пользователя (данную страницу не имеет модуль модератора), код представлен в приложении И;
4. .../ adminquest (adminquest.php) – страница вывода банка вопросов, код представлен в приложении Г;
5. .../adminaddquest (adminaddquest.php) – страница добавления вопросов, код представлен в приложении Д;
6. .../adminchangequest (adminchangequest.php) – страница изменения вопросов, код представлен в приложении Е;
7. .../adminbook (adminbook.php) – страница добавления справочной информации (медицинские учреждения, специальности и группы специальностей) (данную страницу не имеет модуль модератора), код представлен в приложении Л;
7. .../adminaddmedfac (adminaddmedfac.php) – страница добавления медицинских учреждений (данную страницу не имеет модуль модератора), код представлен в приложении М;
8. .../adminaddgroup (adminaddgroup.php) – страница добавления группы специальностей (данную страницу не имеет модуль модератора), код представлен в приложении Н;
9. .../adminaddspecialty (adminaddspecialty.php) – страница добавления специальности (данную страницу не имеет модуль модератора), код представлен в приложении П;
10. .../adminexpected (adminexpected.php) – страница для вывода списка ожидаемых пользователей, код представлен в приложении Р.
11. .../adminaddexpected (adminaddexpected.php) – страница для добавления ожидаемых пользователей, код представлен в приложении С.
12. .../adminreports (adminreports.php) – страница для отчетности результатов тестирования пользователей, код представлен в приложении К.

2.2.3 Описание модуля администратора

Вход в систему осуществляется с главной страницы системы аттестационного тестирования. Для администратора и модератора логин и пароль внесены в БД под соответствующему типу пользователя.

Чтобы войти в модуль администратора нужно перейти по адресу, на котором размещена система. В соответствующие поля необходимо ввести логин и пароль администратора (рисунок 15). В БД определен тип пользователя администратор.

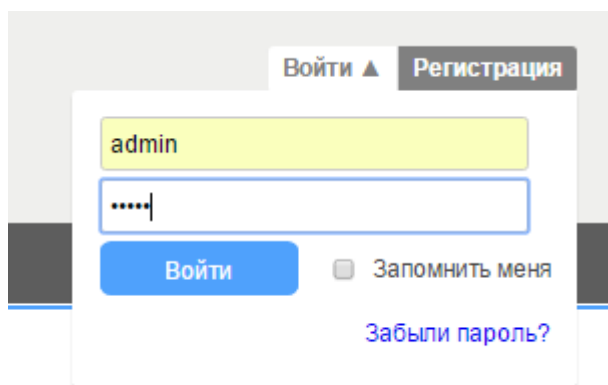


Рисунок 15 – Окно авторизации системы аттестационного тестирования медицинских работников

При определении входа в модуль администратора верхняя часть (шапка) отображаются следующие пункты меню: «Результаты Тестирования», «Пользователи», «Банк Вопросов», «Справочники», «Список Ожидаемых», «Отчеты», «Выйти». Шапка модуля администратора представлена на рисунке 16.

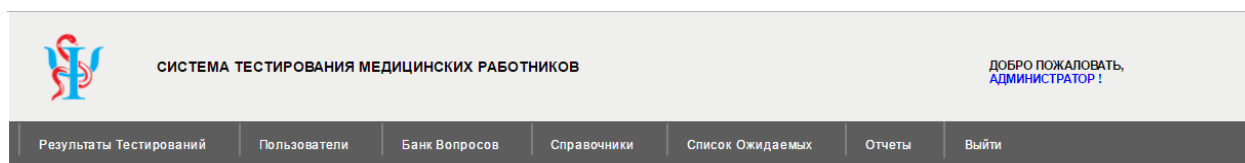


Рисунок 16 – Верхняя часть (шапка) модуля администратора

На главной странице «Результаты Тестирований» модуля администратора, которая представлена на рисунке 17, выгружаются данные о результатах тестирования пользователей в табличном виде с основными данными: ID тестирования, дата тестирования, тип теста, результат, фамилия, имя, отчество, телефон, специальность, место работы. Данные отображаются от самого последнего тестирования (по дате тестирования).

Результаты Тестирований											
Результаты тестирований											
ID	Дата тестирования	Тип теста	Результат	Фамилия	Имя	Отчество	e-mail	Телефон	Специальность	Категория	Место работы
11	2017-06-13 16:07:05	Основной	50/70 (71.4%)	Демченко	Ольга	Николаевна	demchenko@mail.ru	9614508990	Педиатрия	Вторая	ГБУЗ РХ «Таштыпская районная больница»
10	2017-06-12 12:18:07	Пробный	10/20 (50%)	Демченко	Ольга	Николаевна	demchenko@mail.ru	9614508990	Педиатрия	Вторая	ГБУЗ РХ «Таштыпская районная больница»
9	2017-06-08 17:05:02	Основной	51/70 (72.9%)	Майнагашева	Ксения	Ивановна	mainag@mail.ru	9832406657	Акушерство и гинекология	Вторая	ГБУЗ РХ «Аскизская межрайонная больница»
8	2017-06-08 08:06:13	Пробный	20/20 (100%)	Майнагашева	Ксения	Ивановна	mainag@mail.ru	9832406657	Акушерство и гинекология	Вторая	ГБУЗ РХ «Аскизская межрайонная больница»
7	2017-06-06 19:22:36	Основной	48/70 (68.6%)	Чильчигашева	Валерия	Семеновна	Lera5066@mail.ru	9831915066	Сестринское дело	Первая	ГБУЗ РХ «Абаканская городская клиническая поликлиника»
6	2017-06-07 11:54:19	Основной	55/70 (78.6%)	Булдыров	Евгений	Владимирович	buldirov@mail.ru	9134432321	Стоматология	Вторая	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»
4	2017-06-06 17:11:47	Пробный	19/20 (95%)	Чильчигашева	Валерия	Семеновна	Lera5066@mail.ru	9831915066	Сестринское дело	Первая	ГБУЗ РХ «Абаканская городская клиническая поликлиника»
3	2017-06-06 12:30:10	Пробный	15/20 (75%)	Булдыров	Евгений	Владимирович	buldirov@mail.ru	9134432321	Стоматология	Вторая	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»
2	2017-06-04 17:16:12	Основной	59/70 (84.3%)	Карамашева	Татьяна	Петровна	ktp94@mail.ru	9614436013	Детская стоматология	Вторая	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»

Рисунок 17 – Главная страница «Результаты Тестирований» модуля администратора

Страница «Пользователи» доступна по адресу «.../adminusers» и представляет собой перечень всех пользователей системы аттестационного тестирования медицинских работников в табличном виде с основными данными: ID пользователя, фамилия, имя, отчество, телефон, специальность, стаж работы по специальности, место работы, категория, наличие пробного и основного тестирования и блокировка пользователя.

Значения в столбце «Результат» отображает то, что за одну аттестацию у пользователя есть 3 попытки: пробное, основное и дополнительное,

которое дается с письменного распоряжения Министерства здравоохранения Республики Хакасия.

На рисунке 18 представлена страница «Пользователи».

Результаты Тестирований				Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти			
Пользователи												
				ID	Фамилия	Имя	Отчество	Специальность	Стаж работы	Место работы	Категория	e-mail
Изменить	Разблокировать	Обнулить попытку	Добавить попытку	7	Демченко	Ольга	Николаевна	Педиатрия	8	ГБУЗ РХ «Таштыпская районная больница»	Вторая	demchenko@mail.ru
Изменить	Заблокировать	Обнулить попытку	Добавить попытку	6	Майнагашева	Ксения	Ивановна	Акушерство и гинекология	14	ГБУЗ РХ «Аскизская межрайонная больница»	Вторая	mainag@mail.ru
Изменить	Заблокировать	Обнулить попытку	Добавить попытку	5	Чильчигашева	Валерия	Семеновна	Сестринское дело	3	ГБУЗ РХ «Абаканская городская клиническая поликлиника»	Первая	Lera5066@mail.ru
Изменить	Заблокировать	Обнулить попытку	Добавить попытку	4	Булдыров	Евгений	Владимирович	Стоматология	8	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	Вторая	buldirov@mail.ru
Изменить	Заблокировать	Обнулить попытку	Добавить попытку	3	Карамашева	Татьяна	Петровна	Детская стоматология	7	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	Вторая	ktp94@mail.ru
Изменить	Заблокировать	Обнулить	Добавить	2			Молератор	0	0	0	0	

Рисунок 18 – Страница «Пользователи» модуля администратора, лист 1

Результаты Тестирования Пользователи Банк Вопросы Справочники Список Ожидаемых Отчеты Выйти										
Пользователи										
Отчество	Специальность	Стаж работы	Место работы	Категория	e-mail	Телефон	Пробный тест	Основной тест	Дополнительная попытка	Блокировка пользователя
Николаевна	Педиатрия	8	ГБУЗ РХ «Таштыпская районная больница»	Вторая	demchenko@mail.ru	9614508990	1	1	Не предоставлялась	1
Ивановна	Акушерство и гинекология	14	ГБУЗ РХ «Аскизская межрайонная больница»	Вторая	mainag@mail.ru	9832406657	1	1	Не предоставлялась	0
Семеновна	Сестринское дело	3	ГБУЗ РХ «Абаканская городская клиническая поликлиника»	Первая	Lera5066@mail.ru	9831915066	1	1	Предоставлена	0
Владимирович	Стоматология	8	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	Вторая	buldirov@mail.ru	9134432321	1	1	Не предоставлялась	0
Петровна	Детская стоматология	7	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	Вторая	ktp94@mail.ru	9614436013	1	1	Не предоставлялась	0
Молератор	0	0	0	0		0	0	0	Не	0

Рисунок 18, лист 2

Также на странице «Пользователи» присутствует возможности изменения данных пользователей. Для этого нужно нажать на кнопку «Изменить», откроется форма для изменения данных пользователя, которая представлена на рисунке 19 и располагается по адресу .../adminchangeuser.php.

The screenshot shows a web application interface with a top navigation bar containing links: Результаты Тестирований, Пользователи, Банк Вопросов, Справочники, Список Ожидаемых, Отчеты, and Выйти. Below the navigation bar is the title 'Изменить данные пользователя'. The form contains the following fields:

- Фамилия: Демченко
- Имя: Ольга
- Отчество: Николаевна
- Специальность: Педиатрия (dropdown menu)
- Стаж: 8
- Место работы: ГБУЗ РХ «Таштыпская районная больница» (dropdown menu)
- Категория: Вторая (dropdown menu)
- Телефон: 9614508990
- Пробный тест: 1
- Основной тест: 0

A blue button labeled 'Сохранить' (Save) is located at the bottom right of the form. At the bottom of the page, there is a footer with the text '© 2017 ГКУЗ РХ "РМИАЦ"' on the left and 'Техническая поддержка: 8 (3902) 295-025' on the right.

Рисунок 19 – Страница «Изменить данные пользователя»

По мере необходимости на странице «Пользователи», представленной на рисунке 18, имеется возможность заблокировать и разблокировать учетную запись пользователя, перейдя по кнопке «Заблокировать». В поле «Блокировка пользователя» изменится значение с 0 на 1, то есть значение «0» – пользователь не заблокирован, а значение «1» – заблокирован. После блокирования пользователя кнопка «Заблокировать» изменится на «Разблокировать».

Также на странице «Пользователи» имеется функция аннулировать результат тестирования пользователя. Аннулирование результатов тестирования – это дается возможность пройти тестирование при следующем прохождении аттестации. При этом все результаты тестирования пользователя сохраняются.

Дополнительная попытка (вторая после основного) дается только по соответствующему распоряжению Минздрава Хакасии, если предыдущий результат был ниже регламентированного результат 70 %.

Страница «Банк Вопросов» расположена по адресу .../ adminquest. На странице перечень вопросов выгружаются только по определенной специальности. Для просмотра соответствующего перечня вопросов необходимо выбрать специальность при том, что специальности разбиты по группа. На рисунке 20 представлена страница с выгрузкой вопросов и вариантов ответов по специальности.

Вопрос / Варианты ответов	
Изменить	Удалить
Личная дистанция при повседневном общении медицинской сестры с пациентами, коллегами	
0-45см	
45-120см	
120-400см	
400-550см	
Изменить	Удалить
Количество сердечных сокращений в одну минуту у взрослого в норме	
100-120	
90-100	
60-80	
40-60	
Изменить	Удалить
Нормальное артериальное давление- это давление	
Меньше 120/80 мм рт.ст.	
Меньше 130/85 мм рт.ст.	
Больше 130/85 мм рт.ст.	
Больше 140/90 мм рт.ст.	

Рисунок 20 – Страница «Банк Вопросов» модуля администратора

Для работы с банком вопросов существует две функции: изменить и удалить. Отображенные в форме вопросы возможно удалить из БД, нажав на кнопку «Удалить».

Для того чтобы изменить вопрос или варианты ответов, необходимо нажать на кнопку «Изменить». Далее откроется форма для изменения вопроса, расположенная по адресу ... /adminchangequest.php. Страница

«Изменить вопрос» представлена на рисунке 21. После сохранения вопрос обновляется в БД.

Результаты Тестирований | Пользователи | Банк Вопросов | Справочники | Список Ожидаемых | Отчеты | Выйти

Изменить вопрос

Личная дистанция при повседневном общении медицинской сестры с пациентами, коллегами

☐ 0-45см

☐ 45-120см

☐ 120-400см

☐ 400-550см

Сохранить

© 2017 ГКУЗ РК "РМИАЦ" | Техническая поддержка: 8 (3902) 295-025

Рисунок 21 – Страница «Изменить вопрос»

Страница «Добавить вопрос» расположена по адресу .../adminaddquest.php, представленная на рисунке 22. Вопросы можно добавлять как по одному, так и загрузить файл с вопросами с расширением .qst.

Результаты Тестирований | Пользователи | Банк Вопросов | Справочники | Список Ожидаемых | Отчеты | Выйти

Добавить вопрос

Группа специальностей: Средний медицинский персонал

Специальность: Сестринское дело

Выберите файл | Файл не выбран | Добавить из файла

Возбудителями внутрибольничных инфекций могут быть

☐ Бактерии, вирусы

☐ Грибы

☐ Простейшие, многоклеточные паразиты

☒ Все ответы правильные

☐ Введите вариант ответа 5

☐ Введите вариант ответа 6

Сохранить

© 2017 ГКУЗ РК "РМИАЦ" | Техническая поддержка: 8 (3902) 295-025

Рисунок 22 – Страница «Добавить вопрос»

Вопросы в БД добавляются по специальности, поэтому перед тем как добавить вопрос необходимо выбрать специальность. Далее заполнить соответствующие поля. Максимальное число вариантов ответов равно 6, на

против каждого из которых есть checkbox, который необходим для отметки правильных вариантов ответа.

Каждая медицинская структура по определенной специальности предоставляется перечень вопросов в файле с расширением .qst. Пример документа представлен на рисунке 23.

Вопросы между собой разделены знаком «?», а варианты ответов помечены знаками «-» и «+». Соответственно знак «-» обозначает неправильный вариант ответа, а «+» – правильный. Вопрос от вариантов ответов разделяется знаком «:».

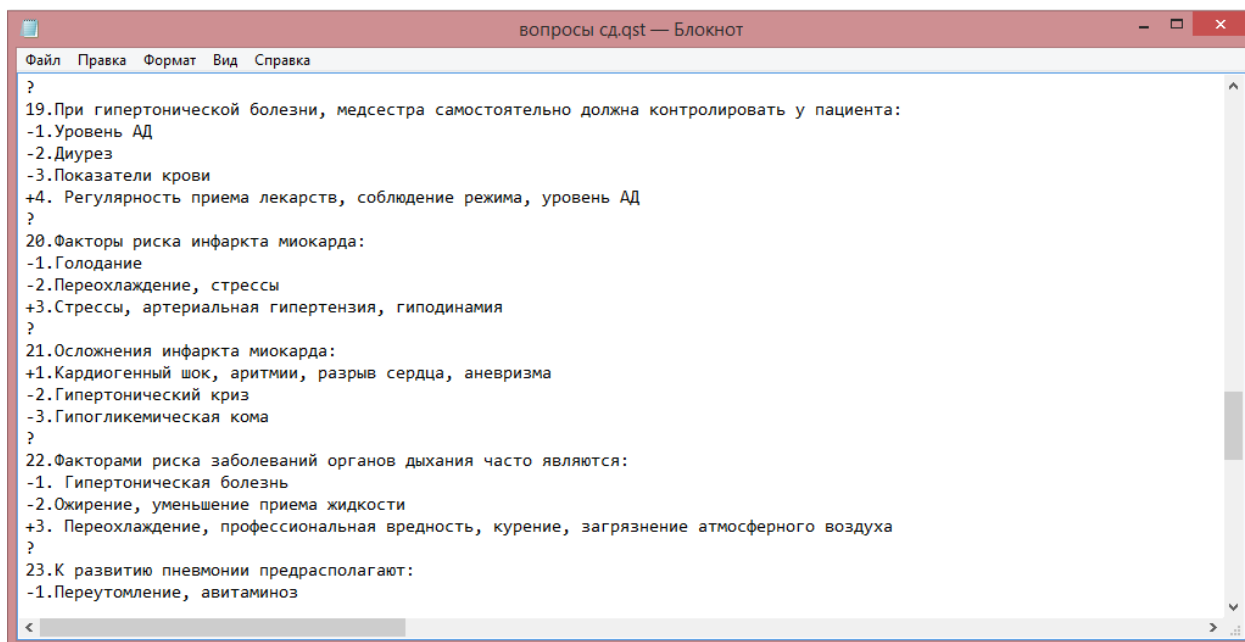


Рисунок 23 – Перечень вопросов в форме для загрузки в БД

Страница «Справочники» расположена по адресу .../adminbook.php. На странице изменяется или добавляется справочная информация: медицинские учреждения, группы специальностей и специальности. На рисунке 24 представлена страница «Справочники».

Результаты Тестирований	Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти
-------------------------	--------------	---------------	-------------	------------------	--------	-------

Справочники

Медицинские учреждения

Добавить

ID	Медицинские учреждения	
1	Хакасская республиканская ассоциация врачей	Удалить
2	ГБУ РХ "Ресфармация"	Удалить
3	ГБУЗ РХ «Боградская районная больница»	Удалить
4	ГБУЗ РХ «Черногорская межрайонная детская больница»	Удалить
5	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	Удалить
6	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	Удалить
7	ГБУЗ РХ «Саяногорская городская детская поликлиника имени Л.Д. Ганичевой»	Удалить
8	ГБУЗ РХ «Саяногорская городская станция скорой медицинской помощи»	Удалить
9	ГБУЗ РХ «Саяногорская городская стоматологическая поликлиника»	Удалить
10	ГБУЗ РХ «Саяногорская межрайонная больница рабочего поселка Майна»	Удалить

Рисунок 24 – Страница «Справочники» модуля администратора, лист 1

25	ГБУЗ РХ «Республиканская клиническая офтальмологическая больница имени Н. М. Одежкина»	Удалить
----	--	---------

Группы специальностей

Добавить

ID	Группа специальностей	
1	Медицинский и фармацевтический персонал	Удалить
2	Средний медицинский персонал	Удалить

Специальности

Добавить

ID	Специальность	Группа специальностей	
1	Стоматолог-ортопед	Медицинский и фармацевтический персонал	Удалить
2	Акушерство и гинекология	Медицинский и фармацевтический персонал	Удалить
3	Аллергология и иммунология	Медицинский и фармацевтический персонал	Удалить
4	Детская стоматология	Медицинский и фармацевтический персонал	Удалить

Рисунок 24, лист 2

В каждый справочник возможно добавить и удалить данные. На рисунке 25 представлена форма для добавления названия медицинского учреждения. Аналогичная форма реализована для добавления названия группы специальности.

Результаты Тестирований	Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти
-------------------------	--------------	---------------	-------------	------------------	--------	-------

Добавить медицинское учреждение

ГБУЗ РХ «Таштыпская районная больница»

Добавить

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 25 – Страница «Добавить медицинское учреждение» модуля администратора

На рисунке 26 представлена форма для добавления названия специальности. Перед вводом названия специальности необходимо выбрать группу, к которой относится специальность.

Результаты Тестирований	Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти
-------------------------	--------------	---------------	-------------	------------------	--------	-------

Добавить специальность

Средний медицинский персонал ▼

Микробиология

Добавить

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 26 – Страница «Добавить специальность» модуля администратора

Страница «Список Ожидаемых» расположена по адресу `.../adminexpected.php`. Список Ожидаемых – это список пользователей, которые ожидаются для тестирования. Если пользователя нет в этом списке, он не сможет войти в систему для прохождения тестирования. На странице выводится список медицинских работников, которые должны пройти тестирование. На рисунке 27 представлена страница «Список Ожидаемых».

Результаты Тестирования	Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти
-------------------------	--------------	---------------	-------------	------------------	--------	-------

Список ожидаемых пользователей для тестирования

Добавить пользователя

	ID	Фамилия	Имя	Отчество	Место работы	Специальность	Дата
Удалить	9	Сулейманова	Наталья	Семеновна	ГБУЗ РХ «Аскизская межрайонная больница»	Сестринское дело	2017-06-08
Удалить	8	Демченко	Ольга	Николаевна	ГБУЗ РХ «Таштыпская районная больница»	Педиатрия	2017-06-08
Удалить	7	Топакова	Наталья	Игоревна	ГБУЗ РХ «Абазинская городская больница»	Педиатрия	2017-06-07
Удалить	6	Владыко	Дмитрий	Геннадьевич	ГБУЗ РХ «Абазинская городская больница»	Акушерство и гинекология	2017-06-06
Удалить	5	Чепсаракова	Анисья	Борисовна	ГБУЗ РХ «Черногорская межрайонная детская больница»	Детская хирургия	2017-06-04
Удалить	4	Чильчигашева	Валерия	Семеновна	ГБУЗ РХ «Абаканская городская клиническая поликлиника»	Сестринское дело	2017-06-04
Удалить	3	Карамашева	Татьяна	Петровна	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	Детская стоматология	2017-06-02
Удалить	2	Булдыров	Евгений	Владимирович	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	Стоматология	2017-06-02
Удалить	1	Майнагашева	Ксения	Ивановна	ГБУЗ РХ «Аскизская межрайонная больница»	Акушерство и гинекология	2017-06-02

Рисунок 27 – Страница «Список Ожидаемых» модуля администратора

На рисунке 28 представлена форма для добавления пользователя для прохождения тестирования. Система идентифицирует пользователя по фамилии, месту работы и специальности.

Результаты Тестирования	Пользователи	Банк Вопросов	Справочники	Список Ожидаемых	Отчеты	Выйти
-------------------------	--------------	---------------	-------------	------------------	--------	-------

Добавить пользователя

Добавить

© 2017 ГКУЗ РХ "РМИАЦ" Техническая поддержка: 8 (3902) 295-025

Рисунок 28 – Страница «Добавить пользователя» модуля администратора

И соответственно для завершения сессии служит кнопка «Выйти».

2.2.4 Описание модуля модератора

Модуль модератора аналогичен модулю администратора. Для входа в модуль модератора вводится логин и пароль модератора, который также внесен в БД под соответствующим типом пользователя.

Страницы модуля администратора аналогичны страницам модуля модератора, которые расположены по одному адресу, за исключением страницы «Справочник», которая отсутствует в модуле модератора.

Для отображения главного меню модератора, которое представлено на рисунке 29, используется проверка сессионной переменной «а» на значение 1 или 2 соответственно на модуль администратора или модератора, код которой представлен в приложении Т. После проверки выгружает соответствующее меню.

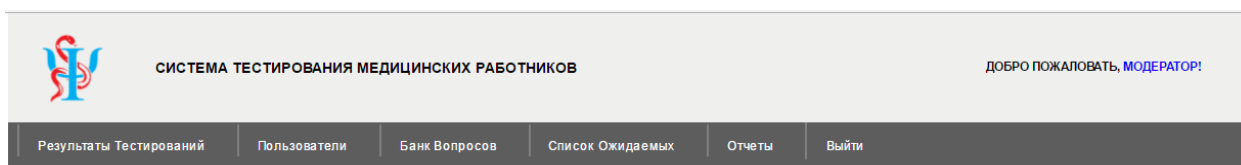


Рисунок 29 – Шапка модуля модератора

На странице «Пользователи» в модуле модератора отсутствуют функции изменения данных пользователей и блокирование пользователя. Страница «Пользователи» представлена на рисунке 30. Проверка на наличие определенных функций проходит также через сессионную переменную, код представлен в приложении Т.

Результаты Тестирований		Пользователи		Банк Вопросов	Список Ожидаемых	Отчеты	Выйти				
Пользователи											
		ID	Фамилия	Имя	Отчество	Специальность	Стаж работы	Место работы	Категория	e-mail	Телефон
Обнулить попытку	Добавить попытку	7	Демченко	Ольга	Николаевна	Педиатрия	8	ГБУЗ РХ «Таштыпская районная больница»	Вторая	demchenko@mail.ru	9614508990
Обнулить попытку	Добавить попытку	6	Майнагашева	Ксения	Ивановна	Акушерство и гинекология	14	ГБУЗ РХ «Аскизская межрайонная больница»	Вторая	mainag@mail.ru	9832406657
Обнулить попытку	Добавить попытку	5	Чильчигашева	Валерия	Семеновна	Сестринское дело	3	ГБУЗ РХ «Абаканская городская клиническая поликлиника»	Первая	Lera5066@mail.ru	9831915066
Обнулить попытку	Добавить попытку	4	Булдыров	Евгений	Владимирович	Стоматология	8	ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	Вторая	buldirov@mail.ru	9134432321
Обнулить попытку	Добавить попытку	3	Карамашева	Татьяна	Петровна	Детская стоматология	7	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	Вторая	ktp94@mail.ru	9614436013
Обнулить	Добавить	2			Модератор	0	0	0	0		0

Рисунок 30 – Страница «Пользователи» модуля модератора

Страницы «Банк вопросов», «Добавить Вопрос» и «Отчеты» аналогичны страницам модуля администратора.

2.2.5 Результат работы модулей администратора и модератора

Результатная информация системы аттестационного тестирования представляет с собой отчеты с отображением в виде таблицы, представленные на рисунке 31:

1. Список пользователей, прошедших тестирование с выборкой по специальности;
2. Список пользователей, не прошедших тестирование с выборкой по специальности;
3. Отчет по количеству пользователей, прошедших и не прошедших тестирование, с выводом по специальности;
4. Список пользователей, прошедших тестирование, с выводом с выборкой по месту работы;

5. Список пользователей, не прошедших тестирование с выборкой по месту работы;

6. Отчет по количеству пользователей, прошедших и не прошедших тестирование, с выводом по месту работы.

Страницы «Отчеты» расположены по адресу .../adminreports.php.

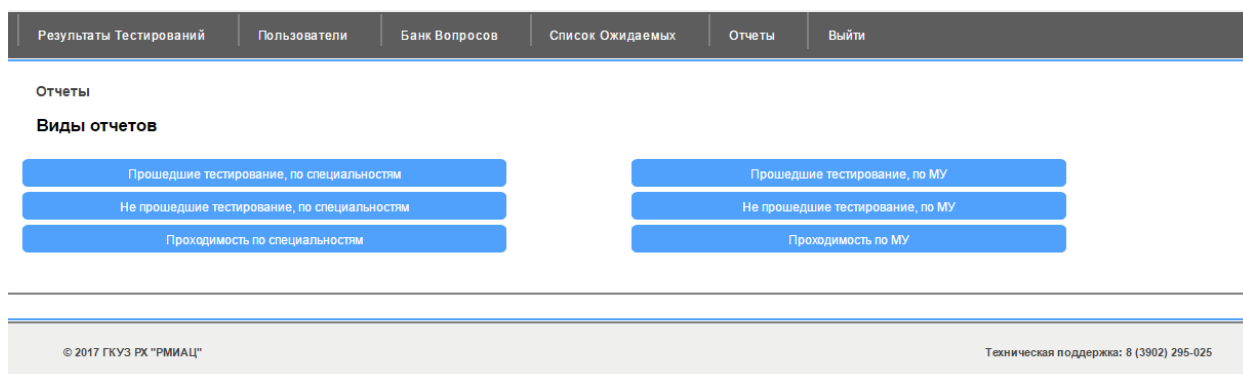


Рисунок 31 – Страница «Отчеты» модуля администратора

Отчет «Прошедшие тестирование, по специальности» позволяет посмотреть результаты тестирования медицинских работников, которые прошли тестирование, по определенной специальности. Также при формировании отчета есть возможность указать необходимый период. На рисунке 32 представлены параметры отчета, по которым будет формироваться.

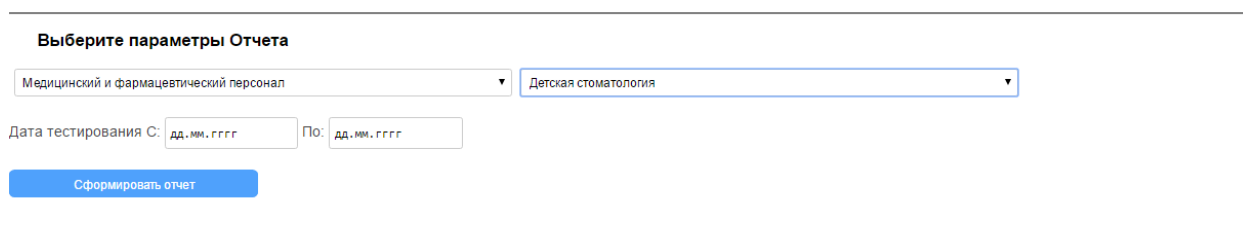


Рисунок 32 – Выборка параметров отчета «Прошедшие тестирование, по специальности»

Отчет представляет собой таблицу с данными медицинских работников. На рисунке 33 представлен отчет по специальности.

Отчет по специальности

Прошедшие тестирование, по специальностям

Специальность: Детская стоматология

Всего записей: 2										
ID	Дата тестирования	Тип теста	Результат	Фамилия	Имя	Отчество	e-mail	Телефон	Специальность	Место работы
1	2017-06-04 12:23:45	Пробный	18/20 (90%)	Карамашева	Татьяна	Петровна	ktp94@mail.ru	9614436013	Детская стоматология	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»
2	2017-06-04 17:16:12	Основной	59/70 (84.3%)	Карамашева	Татьяна	Петровна	ktp94@mail.ru	9614436013	Детская стоматология	ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»

Рисунок 33 – Отчет «Прошедшие тестирование, по специальности»

Отчет «Не прошедшие тестирование, по специальности», представленный на рисунке 34, позволяет посмотреть результаты тестирования медицинских работников, которые не прошли тестирование, по определенной специальности. Выборка отчета аналогична параметрам отчета «Прошедшие тестирование, по специальности», представленной на рисунке 32.

Отчет по специальности

Не прошедшие тестирование, по специальностям

Специальность: Педиатрия

Всего записей: 1										
ID	Дата тестирования	Тип теста	Результат	Фамилия	Имя	Отчество	e-mail	Телефон	Специальность	Место работы
10	2017-06-12 12:18:07	Пробный	10/20 (50%)	Демченко	Ольга	Николаевна	demchenko@mail.ru	9614508990	Педиатрия	ГБУЗ РХ «Таштыпская районная больница»

Рисунок 34 – Отчет «Не прошедшие тестирование, по специальности»

Отчет «Проходимость по специальности», представленный на рисунке 35, позволяет провести анализ по количеству прошедших и не прошедших тестирование по специальностям.

Отчет по количеству прошедших и не прошедших тестирование, по специальностям

Специальность	Количество прошедших	Количество не прошедших	Всего
Акушерство и гинекология	2	0	2
Детская стоматология	2	0	2
Педиатрия	1	1	2
Сестринское дело	1	1	2
Стоматология	2	0	2
Всего:	8	2	10

Рисунок 35 – Отчет «Проходимость по специальности»

Отчет «Прошедшие тестирование, по медицинским учреждениям» позволяет посмотреть результаты тестирования медицинских работников, которые не прошли тестирование, с фильтрацией по медицинским учреждениям и периоду, которая представлена на рисунке 36.

Выберите параметры Отчета

ГБУЗ РХ «Аскизская межрайонная больница» ▼

Дата тестирования С: 01.06.2017 По: 14.06.2017 x ▼

Сформировать отчет

Рисунок 36 – Выборка параметров отчета «Прошедшие тестирование, по медицинским учреждениям»

На рисунке 37 рисунке представлен отчет ««Прошедшие тестирование, по медицинским учреждениям»».

Отчет по медицинским учреждениям

Прошедшие тестирование, по МУ

Медицинское учреждение: ГБУЗ РХ «Аскизская межрайонная больница»

С: 2017-06-01

По: 2017-06-14

Всего записей: 2											
ID	Дата тестирования	Тип теста	Результат	Фамилия	Имя	Отчество	e-mail	Телефон	Специальность	Категория	Место работы
8	2017-06-08 08:06:13	Пробный	20/20 (100%)	Майнагашева	Ксения	Ивановна	mainag@mail.ru	9832406657	Акушерство и гинекология	Вторая	ГБУЗ РХ «Аскизская межрайонная больница»
9	2017-06-08 17:05:02	Основной	51/70 (72.9%)	Майнагашева	Ксения	Ивановна	mainag@mail.ru	9832406657	Акушерство и гинекология	Вторая	ГБУЗ РХ «Аскизская межрайонная больница»

Рисунок 37 – Отчет «Прошедшие тестирование, по медицинским учреждениям»

Отчет «Не прошедшие тестирование, по медицинским учреждениям», представленный на рисунке 38, позволяет посмотреть результаты тестирования медицинских работников, которые не прошли тестирование, по месту своей работы – медицинское учреждение. Выборка отчета аналогична

параметрам отчета «Прошедшие тестирование, по специальности», представленной на рисунке 36.

Отчет по медицинским учреждениям											
Не прошедшие тестирование, по МУ											
Медицинское учреждение: ГБУЗ РХ «Таштыпская районная больница»											
С: 2017-06-01											
По: 2017-06-14											
Всего записей: 1											
ID	Дата тестирования	Тип теста	Результат	Фамилия	Имя	Отчество	e-mail	Телефон	Специальность	Категория	Место работы
10	2017-06-12 12:18:07	Пробный	10/20 (50%)	Демченко	Ольга	Николаевна	demchenko@mail.ru	9614508990	Педиатрия	Вторая	ГБУЗ РХ «Таштыпская районная больница»

Рисунок 38 – Отчет «Не прошедшие тестирование, по медицинским учреждениям»

Отчет «Проходимость по медицинским учреждениям», представленный на рисунке 39, позволяет провести анализ по количеству прошедших и не прошедших тестирование по медицинским учреждениям.

Отчет по количеству прошедших и не прошедших тестирование, по медицинским учреждениям			
Медицинское учреждение	Количество прошедших	Количество не прошедших	Всего
ГБУЗ РХ «Абаканская городская клиническая стоматологическая поликлиника»	2	0	2
ГБУЗ РХ «Саяногорская городская поликлиника рабочего поселка Черемушки»	2	0	2
ГБУЗ РХ «Абаканская городская клиническая поликлиника»	1	1	2
ГБУЗ РХ «Аскизская межрайонная больница»	2	0	2
ГБУЗ РХ «Таштыпская районная больница»	1	1	2
Всего:	8	2	10

Рисунок 39 – Отчет «Проходимость по медицинским учреждениям»

2.2.6 Описание работы системы аттестационного тестирования

Для полноценной работы системы аттестационного тестирования медицинских работников объединяются модули администратора и модератора с модулем тестирования.

Точкой входа системы является файл «index.php», в котором подключаются файлы шапки сайта «header.php» и подвал сайта «footer.php».

На главной странице, представленной на рисунке 40, пользователь входит в систему под логином и паролем. Новый пользователь регистрируется, проходя по соответствующей ссылке. Логин и пароля для администратора и модератора прописывается в БД, в которой в поле «Тип пользователя» существуют соответствующие значения для всех трех пользователей системы. При регистрации данное значение в поле соответствует типу простого пользователя системы.

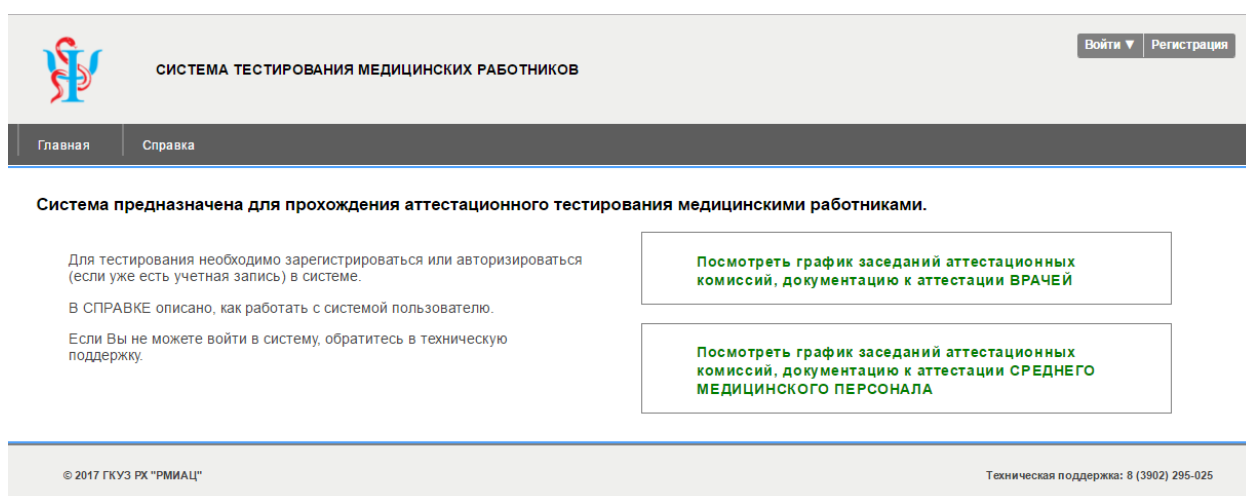


Рисунок 40 – Главная страница системы аттестационного тестирования

То есть по соответствующему типу пользователя система отображает необходимые страницы и данные.

2.2.7 Обеспечение информационной безопасности

В системе аттестационного тестирования медицинских работников обрабатываются персональные данные пользователей, которые вводятся при регистрации, поэтому необходимо обеспечить информационную безопасность, в частности обеспечить следующую защиту:

1. защита от прямого внедрения вредоносных инструкций в SQL-запросы;

2. шифрованное хранение паролей и ссылки на подтверждение электронной почты;
3. разграничение прав доступа;
4. обеспечение безопасного и конфиденциального обмена информацией между веб-сайтом и устройством.

Прямое внедрение вредоносных инструкций в SQL-запросы – это методика, в которой взломщик создает или изменяет текущие SQL-запросы для отображения скрытых данных, их изменения или даже выполнения опасных команд операционной системы на сервере базы данных. Атака выполняется на базе приложения, строящего SQL-запросы из пользовательского ввода и статических параметров. Благодаря отсутствию проверки пользовательского ввода и соединению с базой данных под учетной записью суперпользователя (или любого другого пользователя, наделенного соответствующими привилегиями), взломщик может создать еще одного пользователя БД с правами суперпользователя [15]. Перед отправкой введенных пользователем данных преобразуются любые нечисловые данные специальной экранирующей функцией (`mysqli_real_escape_string`, `trim`). Производится проверка содержания всех полей форм средствами HTML5.

В БД хранятся пароли, сессионные переменные и ссылки на подтверждение электронной почты, перед отправкой которых на сервер указанные данные зашифровывает однонаправленным шифром md5, алгоритм которого зашифровывает данные в формате 128-bit hash (контрольную сумму) и используется для проверки подлинности данных, когда происходит их передача в зашифрованном виде. Такой подход обеспечивает значительное снижение риска взлома, а значит и высокую защиту личных данных пользователей.

Существует разграничение прав доступа. Изменение таблицы «users» возможно при получении доступа в БД, например, через интерфейс PhpMyAdmin. То есть для доступа к модулю администратора или модератора пользователю необходимо выполнить авторизацию в модуле администратора

или модератора путем ввода, известного только данному человеку, пароля. После чего введенный пароль преобразуется в специальный хеш-код и сравнивается с хеш-кодом в таблице «users».

Для обеспечения безопасного и конфиденциального обмена информацией между веб-сайтом и устройством, которое использует пользователь для доступа в сеть используется протокол передачи данных https, который шифрует всю передаваемую информацию, фиксирует любые изменения или искажения передаваемых данных, гарантирует безопасную аутентификацию пользователя.

Вывод по проектному разделу

Для отображения функционала модулей администратора и модератора построена диаграмма потоков данных DFD, где определены входные и выходные данные и взаимосвязь между данными и набором функций модулей, на основании которой выявлены таблицы и справочники, также связь между таблицами. База данных реализована СУБД MySQL.

Программный код функционала модулей администратора и модератора написан на объектно-ориентированном языке программирования PHP, языке разметки гипертекста HTML с помощью таблицы стилей CSS и языка JavaScript.

Результатом работы модулей администратора и модератора являются отчеты.

Модули администратора и модератора интегрированы с модулем тестирования в единую систему аттестационного тестирования медицинских работников. Система интегрирована через авторизацию пользователей. На главной странице системы присутствует окно авторизации и регистрации, где, вводя соответствующий логин и пароль, пользователь входит в систему, но также есть логин и пароль соответственно для администратора и модератора, под которыми они входят каждый в соответствующий модуль.

Разработанная система аттестационного тестирования медицинских работников позволяет автоматизировать процесс аттестационного тестирования медицинских работников на уровне республики Хакасия.

3 Оценка экономической эффективности внедрения информационной системы аттестационного тестирования медицинских работников в процесс аттестации

3.1 Расчет эксплуатационных затрат

Для поддержания надлежащего технического состояния системы необходимо вкладывать в ее эксплуатацию денежные средства, поэтому рассчитываются эксплуатационные затраты, которые, в отличие от капитальных, являются повторяющимися, рассчитываются в сумме за год и составляют себестоимость продукции или услуг.

Эксплуатационные затраты рассчитываются по следующей формуле:

$$C = C_{\text{ЗП}} + C_{\text{АО}} + C_{\text{ТО}} + C_{\text{ЛС}} + C_{\text{НИ}} + C_{\text{ПРОЧ}}, \quad (1)$$

где $C_{\text{ЗП}}$ – зарплата управленческого персонала, работающего с использованием ИС (пользователей ИС);

$C_{\text{АО}}$ – амортизационные отчисления;

$C_{\text{ТО}}$ – затраты на техническое обслуживание, включая заработную плату персонала ИС;

$C_{\text{ЛС}}$ – затраты, связанные с использованием глобальных вычислительных сетей;

$C_{\text{НИ}}$ – затраты на носители информации;

$C_{\text{ПРОЧ}}$ – прочие затраты.

С системой аттестационного тестирования медицинских работников управляют следующие пользователи: администратор (сотрудник ГКУЗ РХ

«РМИАЦ») и кадровый работник (сотрудник Минздрава Хакасии). Зарплата администратора составляет 20 500 рублей, а зарплата кадровика – 25 000 рублей.

Для расчета зарплаты именно за работу со системой аттестационного тестирования медицинских работников по следующей формуле:

$$C_{\text{ЗПА}} = \frac{\text{ЗП}}{\text{Д}} * D_A, \quad (2)$$

где $C_{\text{ЗПА}}$ – зарплата управленческого персонала за работу со системой аттестационного тестирования медицинских работников;

ЗП – зарплата управленческого персонала за месяц;

Д – количество рабочих дней управленческого персонала в месяц;

D_A – количество часов работы со системой аттестационного тестирования медицинских работников.

Время работы администратора со системой аттестационного тестирования медицинских работников приблизительно 1 час в день и 21 рабочих часов в месяц, следовательно, за месяц с системой работают приблизительно 21 час. Рабочий день составляет 8 часов. Разделив время работы с системой на рабочие часы, получается, что на систему администратор тратит 2,63 дня в месяц.

Поставляя значения в формулу (2), получается, что зарплата администратора за работу именно со системой аттестационного тестирования медицинских работников составляет 2 600 рублей в месяц и 31 200 рублей в год.

Аналогично рассчитывается зарплату кадрового работника. Кадровый работник работает в системе аттестационного тестирования 3,75 дня, исходя из того, что приблизительно количество проводимых аттестаций равно шести в месяц и время, затрачиваемое на работу с системой, равно 5 часам на 1 аттестацию. Следовательно, его зарплата за работу именно со системой

аттестационного тестирования медицинских работников составляет 4 500 рублей в месяц и 54 000 рублей в год.

Итак, зарплата управленческого персонала, работающего со системой аттестационного тестирования медицинских работников, равна 85 200 рублей в год.

Амортизационные отчисления – это денежные средства, предназначенные для износа объектов.

Сумма годовых амортизационных отчислений определяется по следующей формуле:

$$A_r = C_б * H_{ам}, \quad (3)$$

где A_r – сумма годовых амортизационных отчислений;

$C_б$ – балансовая стоимость оборудования;

$H_{ам}$ – норма амортизации, которая составляет 25 %.

Балансовая стоимость персонального компьютера составляет 35 000 рублей. Срок службы – 5 лет.

Расчет суммы амортизационных отчислений по формуле 3:

$$A_r = 35\,000 * 0,25 = 8\,750 \text{ р.}$$

Так как со системой работают администратор и кадровый работник, следовательно, персонального компьютера в наличии два.

Сумма годовых амортизационных отчислений равна 17 500 рублей.

Затраты на техническое обслуживание: 400 рублей за одно обслуживание персонального компьютера. Зарплата персонала ИС в месяц – 15 000 рублей. В месяц необходимо хоть 1 раз проводить техническое обслуживание. Следовательно, одно техническое обслуживание всего оборудования составит 1 340 рублей. В итоге за 1 год затраты составят – 16 030 рублей.

Затраты, связанные с использованием глобальных вычислительных сетей, составляют 350 рублей в месяц.

Затраты на носители информации: бумага для печати отчетов – 1 пачка в месяц (500 листов), стоимость которой составляет 270 рублей, за год 3240 рублей.

В прочие затраты входят небольшие затраты, поэтому прочие затраты равны 3% от всей суммы эксплуатационных затрат.

В таблице 20 приведен итог по эксплуатационным затратам за 1 год. На рисунке 41 представлен итог в виде сегментированного графика в процентах.

Таблица 20 – Итоговая таблица эксплуатационных затрат

	Состав затрат	Планируемая сумма, руб	Итог, руб
Зарплата управленческого персонала, работающего с использованием ИС (пользователей ИС)	Зарплата администратора	31 200	85 200
	Зарплата кадрового сотрудника	54 000	
Амортизационные отчисления	Персональный компьютер – 2 шт	17 500	17 500
Затраты на техническое обслуживание	Техническое обслуживание 2х ПК	9 600	16 100
	Зарплата персонала ИС	6 500	
Затраты, связанные с использованием глобальных вычислительных сетей	Интернет	4 200	4 200
Затраты на носители информации	Бумага	3240	3 240
Прочие затраты	Звонки, электроэнергия, ремонт ПК	3790	3790
Итог			130 030

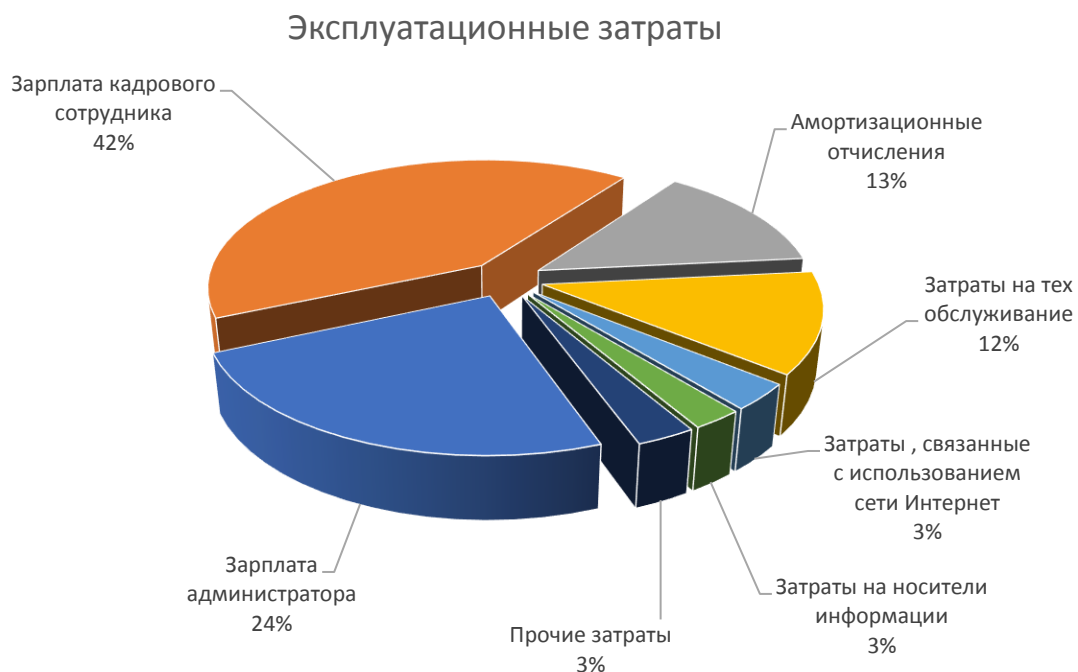


Рисунок 41 – Годовые эксплуатационные затраты

Наибольший удельный вес в эксплуатационных затратах принадлежит заработной плате управленческого персонала, который работает в системе аттестационного тестирования медицинских работников, амортизационным отчислениям и техническому обслуживанию.

3.2 Оценка эффективности проектов в зависимости от технических, технологических, отраслевых или региональных особенностей

В данной ВКР оценка эффективности проекта в зависимости от технических, технологических отраслевых или региональных особенностей не рассматривается, так как перечисленные особенности отсутствуют.

Рассматриваются принципы оценки эффективности проекта независимо от их технических, технологических отраслевых или региональных особенностей.

Первый принцип – рассмотрение проекта на протяжении всего жизненного цикла (расчетного периода) – от проведения исследований до прекращения проекта.

Жизненный цикл системы аттестационного тестирования медицинских работников:

1. исследование процесса аттестации и сбор информации о разработке системы;
2. проектирование системы аттестационного тестирования медицинских работников
3. разработка системы аттестационного тестирования медицинских работников;
4. внедрение системы аттестационного тестирования медицинских работников
5. тестирование работы системы аттестационного тестирования медицинских работников;
6. эксплуатация системы аттестационного тестирования медицинских работников.

Этапы с 1 по 5 объединим в один этап «Разработка системы аттестационного тестирования медицинских работников», так как по временному периоду имеют небольшой промежуток 2 месяца.

Вторым принципом является моделирование денежных потоков, включающих все связанные с осуществлением проекта денежные поступления.

Первый этап – разработка системы аттестационного тестирования медицинских работников. Капитальные затраты данного этапа составляют 65 860 рублей.

Для второго этапа рассчитаны в пункте 3.1 эксплуатационные затраты, которые составляют 130 030 рублей. При расчете будущих эксплуатационных затрат прибавляется 10% в год с учетом возможного финансирования доработок системы и прочих затрат.

На рисунке 42 представлено прогнозирование денежных потоков за период жизненного цикла системы аттестационного тестирования медицинских работников.

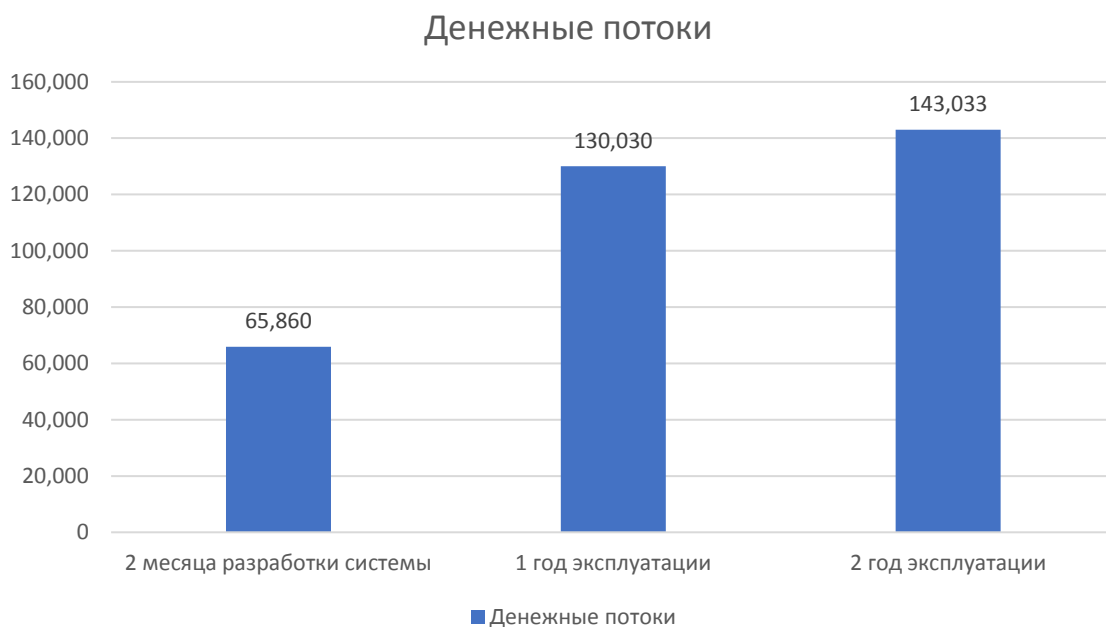


Рисунок 42 – Прогнозирование денежных потоков

Общая сумма на протяжении всего жизненного цикла проекта составляет примерно 338 923 рубля.

3.3 Оценка рисков проекта разработки системы аттестационного тестирования медицинских работников

Процесс управления рисками является актуальным и необходимым для реализации успешных ИТ-проектов.

Возможные риски проекта разработки аттестационного тестирования медицинских работников отмечены следующим списком:

1. увеличение стоимости разработки до 10 %;
2. срыв сроков выплаты денежных средств в период разработки и эксплуатации;

3. срыв сроков проекта;
4. риск потери данных проекта;
5. низкая степень готовности персонала к использованию нового модуля модератора.
6. несовместимость с используемым ПО;
7. срыв сроков опытной эксплуатации в организации;
8. пропуск ошибок при тестировании ИС;
9. отсутствие единой методологии и планов внедрения.

Для того чтобы проект был успешен, нужно определить степень влияния всех возможных рисков проекта. Поэтому используется качественная оценка рисков, которая регламентируется по трехуровневой шкале влияния рисков на реализацию проекта:

- «низкий уровень» – малый риск, не значительное влияние на проект;
- «средний уровень» – риск, имеющий равную вероятность повлиять и не повлиять на результат исполнения проекта;
- «высокий уровень» – высокий риск, заключающийся в неисполнении проекта или в значительном изменении результата его реализации.

В таблице 21 приведен итог по оценке возможных рисков проекта, где так же спланированы мероприятия по снижению рисков, имеющих наивысшую оценку.

Таблица 21 – Перечень возможных рисков проекта

№п.	Перечень рисков проекта	Уровень влияния риска на проект	Возможность предотвращения или снижения риска
1	Увеличение стоимости разработки до 10 %	«Средний уровень»	Заключение договора с фиксированными условиями
2	Срыв сроков выплаты денежных средств	«Низкий уровень»	Заключить четкие сроки выплаты инвестиций
3	Срыв сроков проекта	«Средний уровень»	Усиление контроля над исполнением этапов проекта в заданные сроки, Оценка промежуточных результатов заказчиком, увеличение производительности за счет снижения длительности
4	Риск потери данных	«Средний уровень»	Резервное копирование данных,

	проекта		использование сервисов облачного хранения
5	Низкая степень готовности персонала к использованию нового модуля модератора.	«Высокий уровень»	Показать дальнейшие перспективы работы с системой аттестационного тестирования
6	Несовместимость с используемым ПО	«Средний уровень»	Предусмотреть максимальные сроки тестирования системы
7	Срыв сроков опытной эксплуатации в организации	«Средний уровень»	Установить сроки опытной эксплуатации, согласованные с руководством организации
8	Пропуск ошибок при тестировании ИС	«Высокий уровень»	Увеличить сроки тестирования
9	Отсутствие единой методологии и планов внедрения	«Средний уровень»	Составить план и сроки внедрения системы

Самый высокий риск касается кадрового сотрудника Министерства здравоохранения Республики Хакасия, так как в системе аттестационного тестирования, которую использует Министерство здравоохранения Республики Хакасия в настоящее время, не предусмотрен модуль работы этого сотрудника, что с новой системой аттестационного тестирования основные функции работы будут перенаправлены на кадрового сотрудника. Для снижения данного риска необходимо показать дальнейшие перспективы работы с самой системой аттестационного тестирования, а при необходимости узнать результаты тестирования у администратора системы.

Вывод по экономическому разделу

Капитальные затраты составляют 65 860 рублей на разработку системы аттестационного тестирования медицинских работников, срок которого равен 2 месяцам.

Затраты на эксплуатацию за первый год эксплуатации составляют 130 030 рублей. С учетом инфляции и возможной доработки функционала системы за второй год эксплуатации составляют 143 030 рублей.

Общие затраты на протяжении всего жизненного цикла проекта составляют примерно 338 923 рубля.

Экономическая эффективность внедрения разработанной системы аттестационного тестирования медицинских работников состоит в том, что общие затраты на систему меньше затрат, если бы совершилась покупка готового решения, так как к эксплуатационным затратам прибавилась ежегодная стоимость существующей системы.

ЗАКЛЮЧЕНИЕ

В разделе «Теоретический анализ процесса аттестации медицинских работников Республики Хакасия» проанализирован процесс аттестации медицинских работников, проводимой Министерством здравоохранения Республики Хакасия. На основе анализа построена функциональная модель, отражающая весь процесс.

В распоряжении Министерства здравоохранения Республики Хакасия в настоящее время находится система «Система тестирования медицинских работников 2.1», но система имеет критические ошибки и не отвечает требуемому от заказчика функционалу, поэтому появилась необходимость в новой системе аттестационного тестирования.

Проанализирован рынок подобных систем тестирования: MyTestXPro, x-TLS, INDIGO, Moodle, OpenTest и Let's test. В результате анализа выявлено то, что существующие системы тестирования либо не отвечают требованиям функционала, либо имеют большой набор функционала, который не будет использоваться. Исходя из анализа, решено разрабатывать систему аттестационного тестирования медицинских работников.

В результате проведенного анализа средств разработки выбран язык программирования PHP который имеет возможность свободы выбора операционной системы, веб-сервера и базы данных, удобен в работе с запросами SQL, имеет высокую скорость выполнения и богатую функциональность, и языки, как HTML, JavaScript, CSS. Для проектирования БД выбрана СУБД MySQL, так как она не требует денежных затрат и является клиент-серверной системой.

В разделе «Практическая реализация разработки информационной системы аттестационного тестирования медицинских работников (модули администратора и модератора)» разработана система аттестационного тестирования медицинских работников средствами веб-программирования.

Результатная информация представляется отчетами: отчет по специальности (прошедшие или не прошедшие тестирование), отчет по медицинским учреждениям (прошедшие или не прошедшие тестирование), также анализ по количеству прошедших и не прошедших тестирование по специальностям и медицинским учреждениям.

Цель и задачи ВКР достигнуты.

Разработанная система позволяет автоматизировать процесс аттестационного тестирования медицинских работников на уровне республики Хакасия.

Для развития системы аттестационного тестирования медицинских работников доработать функцию экспорта отчетов в excel-файл.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Об утверждении Положения о Министерстве здравоохранения Республики Хакасия [Электронный ресурс] : Постановление Правительства Республики Хакасия от 11.06.2009 № 260 (в ред. Постановлений Правительства Республики Хакасия от 13.07.2010 № 358, от 28.04.2011 № 225, от 05.05.2012 № 285, от 09.07.2013 № 372, от 27.03.2014 № 138, от 11.12.2014 № 667, от 06.10.2015 № 504, от 08.12.2015 № 648, от 06.09.2016 № 433) // Справочная правовая система «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru>.

2. О Порядке и сроках прохождения медицинскими работниками и фармацевтическими работниками аттестации для получения квалификационной категории [Электронный ресурс] : Приказ Минздрава России от 23.04.2013 № 240н (Зарегистрировано в Минюсте России 05.07.2013 № 29005) // Справочная правовая система «КонсультантПлюс». – Режим доступа: <http://www.consultant.ru>.

3. Министерство здравоохранения Республики Хакасия [Электронный ресурс]. – Режим доступа: <http://mz19.ru>.

4. Ястребова, Е. А. Научное обоснование оптимизации системы аттестации медицинских работников стоматологического профиля на региональном уровне : дис. ... канд. мед. наук : 14.01.14 / Ястребова Елена Александровна. – Тверь, 2010. – 226 с.

5. Корпоративный сайт компании «Корпоративные системы Интернет» [Электронный ресурс] : IT и управленческий консалтинг // IDEF0 – стандарт и методология функционального моделирования. – Москва, 2006. – Режим доступа: <http://corpsite.ru>.

6. Ананченко, И. В. Классификация компьютерных систем тестирования знаний учащихся / И. В. Ананченко // Международный журнал экспериментального образования. – 2016. – № 4-2. – С. 210-213. URL: <http://www.expeducation.ru>.

7. Национальный открытый институт «Интуит» [Электронный ресурс] : Курсы // Лекция 1 Принципы работы и структура Web-приложений на основе ASP.NET. – Режим доступа: <http://www.intuit.ru>.
8. Справочное online руководство по программе MyTestXPro [Электронный ресурс]. – Режим доступа: <http://mytest.klyaksa.net>.
9. Автоматизированная система обучения и контроля знаний на основе расширенных тестовых заданий x-TLS [Электронный ресурс]. – Режим доступа: <http://xtls.org.ua>.
10. Программа для тестов и онлайн тестирований INDIGO [Электронный ресурс]. – Режим доступа: <http://indigotech.ru>.
11. Веб-приложения для бизнеса и образования «Открытые технологии» [Электронный ресурс] : Система дистанционного обучения Русский Moodle 3k. – Режим доступа: <https://www.opentechnology.ru>.
12. Система тестирования STARTEXAM [Электронный ресурс] : Онлайн система тестирования OpenTest. – Режим доступа: <http://www.opentest.ru>.
13. Система тестирования Let's test [Электронный ресурс]. – Режим доступа: <https://letstest.ru/>
14. Типичный программист Tproger [Электронный ресурс] : SQLite, MySQL и PostgreSQL: сравниваем популярные реляционные СУБД. – Режим доступа: <https://tproger.ru>.
15. PHP [Электронный ресурс] : Возможности PHP. – Режим доступа: <http://php.net>.
16. Сайтостроение от А до Я [Электронный ресурс] : PHP, Ruby, Python – краткая характеристика трёх языков программирования. – Режим доступа: <http://www.internet-technologies.ru>.
17. Ruby [Электронный ресурс]. – Режим доступа: <https://www.ruby-lang.org/ru>.
18. Python 3 для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonworld.ru>.

19. Государство. Бизнес. ИТ [Электронный ресурс] : Microsoft SQL Server. – Режим доступа: <http://www.tadviser.ru>.

20. Веб-библиотека [Электронный ресурс] : HTML/ CSS / JavaScript. – Режим доступа: <http://www.weblibrary.biz>.

21. Устав государственного казенного учреждения здравоохранения республики Хакасия "Республиканский медицинский информационно-аналитический центр" (в ред. Постановлений Правительства республики Хакасия от 16.02.2011 № 64, от 06.12.2011 ; 837, от 28.05.2014 ; 237, от 26.08.2014 ; 430) : Постановление Правительства Республики Хакасия от 31.08.2012 № 585. – Введ. 31.08.2012 г. – Абакан, 2012. – 7с.

22. e-educ [Электронный ресурс] : Диаграмма потоков данных (DFD). – Режим доступа: <http://e-educ.ru/bd14.html>.

23. Ковалёв, В. П. Использование информационных технологий в совершенствовании медицинского обслуживания населения : дис. ... канд. мед. наук : 14.02.12 / Ковалёв Владимир Петрович. – Москва, 2012. – 203 с.

ПРИЛОЖЕНИЕ А

Программный код файла «config.php»

```
<?php
define("HOST", "localhost");
define("USER", "root");
define("PASS", "root6104");
define("DB", "testing");

$db = @mysqli_connect(HOST, USER, PASS, DB) or die('Нет соединения с БД');
mysqli_set_charset($db, 'utf8') or die('Не установлена кодировка соединения');
?>
```

ПРИЛОЖЕНИЕ Б

Листинг вывода результатов тестирования файла adminresult.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}
if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}
$spec = get_spec();
$mf = get_facility();
$categ = get_category();
$stab = all_result();

foreach ($stab as $k => $v) {
    foreach ($spec as $s) {
        if ($s['id'] == $v['specialty_t']) {
            $stab[$k]['specialty_t'] = $s['specialty'];
        }
    }
}

foreach ($stab as $k => $v) {
    foreach ($mf as $m) {
        if ($m['id'] == $v['mf']) {
            $stab[$k]['mf'] = $m['med_facility'];
        }
    }
}

foreach ($stab as $k => $v) {
    foreach ($categ as $category) {
        if ($category['id'] == $v['category_t']) {
            $stab[$k]['category_t'] = $category['category'];
        }
    }
}

?>
```

```

<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Результаты тестирований</h2>
<div id="resulttest">
    <table class="resulttest">
        <tr style="font-weight: bold;">
            <td>ID</td>
            <td>Дата тестирования</td>
            <td>Тип теста</td>
            <td>Результат</td>
            <td>Фамилия</td>
            <td>Имя</td>
            <td>Отчество</td>
            <td>e-mail</td>
            <td>Телефон</td>
            <td>Специальность</td>
            <td>Категория</td>
            <td>Место работы</td>
        </tr>
        <?php foreach ($tab as $table): ?>
            <tr>
                <td><?=$table['id']?></td>
                <td><?=$table['date']?></td>
                <td><?=$table['type_test']?></td>
                <td><?=$table['result'] . ' ('.$table['percent'] . '%)'?></td>
                <td><?=$table['surname']?></td>
                <td><?=$table['name']?></td>
                <td><?=$table['patronymic']?></td>
                <td><?=$table['email']?></td>
                <td><?=$table['phone']?></td>
                <td><?=$table['specialty_t']?></td>
                <td><?=$table['category_t']?></td>
                <td><?=$table['mf']?></td>
            </tr>
        <?php endforeach ?>
    </table>
</div>
</form>

<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ В

Листинг вывода всех пользователей файла adminusers.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}
if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$spec = get_spec();
$mf = get_facility();
$categ = get_category();
$stab = all_users();

foreach ($stab as $k => $v) {
    foreach ($spec as $s) {
        if ($s['id'] == $v['specialty']) {
            $stab[$k]['specialty'] = $s['specialty'];
        }
    }
}

foreach ($stab as $k => $v) {
    foreach ($mf as $m) {
        if ($m['id'] == $v['mf']) {
            $stab[$k]['mf'] = $m['med_facility'];
        }
    }
}

foreach ($stab as $k => $v) {
    foreach ($categ as $category) {
        if ($category['id'] == $v['category']) {
            $stab[$k]['category'] = $category['category'];
        }
    }
}
```

```

if (isset($_POST['change'])) {
    $_SESSION['change'] = $_POST['change'];
    header("Location: adminchangeuser.php");
}

if (isset($_POST['block'])) {
    $msg = block_user($_POST);

    if ($msg === TRUE) {
        $_SESSION['msg'] = "Пользователь заблокирован";
        header("Location:".$_SERVER['PHP_SELF']);
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}

if (isset($_POST['unblock'])) {
    $msg = unblock_user($_POST);

    if ($msg === TRUE) {
        $_SESSION['msg'] = "Пользователь разблокирован";
        header("Location:".$_SERVER['PHP_SELF']);
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}

if (isset($_POST['null'])) {
    $msg = null_user($_POST);

    if ($msg === TRUE) {
        $_SESSION['msg'] = "Попытки тестирования пользователя обнулены";
        header("Location:".$_SERVER['PHP_SELF']);
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
}??>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Пользователи</h2>
<div id="resulttest">
    <table class="resulttest">
        <tr style="font-weight: bold;">
            <td style="border-right: none;"></td>
            <td style="border-right: none; border-left: none;"></td>
            <td style="border-left: none;"></td>
            <td>ID</td>

```

```

        <td>Фамилия</td>
        <td>Имя</td>
        <td>Отчество</td>
        <td>e-mail</td>
        <td>Телефон</td>
        <td>Специальность</td>
        <td>Стаж</td>
        <td>Место работы</td>
        <td>Категория</td>
        <td>Согласие на обработку ПДн</td>
        <td>Пробный тест</td>
        <td>Основной тест</td>
        <td>Блокировка пользователя</td>
    </tr>
    <?php foreach ($tab as $table): ?>
        <tr>
            <?php if (isset($_SESSION['a']) && isset($_SESSION['sess']) && ($_SESSION['a'] == 1)): ?>
                <td><button value="<?=$table['user_id']?>" name="change">Изменить</button></td>
                <?php if ($table['block'] == 0) : ?>
            <td><button value="<?=$table['user_id']?>" name="block">Заблокировать</button></td>
                <?php endif; ?>
            <?php if ($table['block'] == 1) : ?>
            <td><button value="<?=$table['user_id']?>" name="unblock">Разблокировать</button></td>
                <?php endif; ?>
                <?php endif; ?>
            <?php if (isset($_SESSION['a']) && isset($_SESSION['sess']) && (($_SESSION['a'] == 1) or
            ($_SESSION['a'] == 2))): ?>
                <td><button value="<?=$table['user_id']?>" name="null">Обнулить попытки</button></td>
                <?php endif; ?>
                <td><?=$table['user_id']?></td>
                <td><?=$table['surname']?></td>
                <td><?=$table['name']?></td>
                <td><?=$table['patronymic']?></td>
                <td><?=$table['email']?></td>
                <td><?=$table['phone']?></td>
                <td><?=$table['specialty']?></td>
                <td><?=$table['experience']?></td>
                <td><?=$table['mf']?></td>
                <td><?=$table['category']?></td>
                <td><?=$table['consent']?></td>
                <td><?=$table['probe_test']?></td>
                <td><?=$table['main_test']?></td>
                <td><?=$table['block']?></td>
            </tr>
        <?php endforeach ?>
    </table>
</div>
</form>
<?php require_once 'footer.php'; ?>

```


ПРИЛОЖЕНИЕ Г

Листинг вывода банка вопросов файла `adminquest.php`

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$groups = get_group();
$category = get_category();

if (isset($_POST['special'])) {
    $tab = all_test($_POST['special']);
}

if (isset($_POST['add'])) {
    header("Location: adminaddquest.php");
}

if (isset($_POST['change'])) {
    $_SESSION['change_quest'] = $_POST['change'];
    header("Location: adminchangequest.php");
}

if (isset($_POST['unset'])) {
    $msg = unset_quest($_POST);

    if ($msg === TRUE) {
        $_SESSION['msg'] = "Вопрос удален";
        header("Location: " . $_SERVER['PHP_SELF']);
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">
```

```

<h2>Банк Вопросов</h2>
<label>Группа специальностей</label>
<select class="form-control" name="group" id="group" style="width: 250px">
  <option disabled selected>Выберите группу</option>
  <?php foreach($groups as $group): ?>
    <option value="<?=$group['id']?>"><?=$group['group']?></option>
  <?php endforeach; ?>
</select>
<label>Специальность</label>
<select class="form-control" name="special" id="special" style="width: 250px">
  <option disabled selected>Выберите специальность</option>
</select>
<input type="submit" name="view" value="Просмотреть">
<div id="resulttest">
  <table class="resulttest">
    <?php if (isset($_POST['special'])): ?>
      <?php if (isset($tab)): ?>
        <tr style="font-weight: bold;">
          <td style="border: none;"></td>
          <td style="border: none;">Вопрос / Варианты ответов</td>
        </tr>
        <?php foreach ($tab as $k => $v): ?>
          <?php foreach ($v as $id_answer => $answer):?>
            <tr>
              <?php if (!$id_answer):?>
                <td style="border: none;"><button value="<?=$k?>" name="change">Изменить</button></td>
                <td style="border: none;"><button value="<?=$k?>" name="unset">Удалить</button></td>
                <td style="border: none;" class=" "><?=$answer?></td>
              <?php else: ?>
                <td style="border: none;"></td>
                <td style="border: none;"></td>
                <td style="border: none;" class=" "><?=$answer?></td>
              <?php endif; ?>
            </tr>
          <?php endforeach ?>
        <?php endforeach ?>
        <?php else: ?>
          <tr>
            <td>Вопросов выбранной специальности нет</td>
          </tr>
        <?php endif; ?>
      <?php else: ?>
        <tr>
          <td>Выберите специальность</td>
        </tr>
      <?php endif; ?>
    </table>
  </div>
</form>
<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ Д

Листинг добавления вопросов файла adminaddquest.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$groups = get_group();
$category = get_category();

if (isset($_POST['addquest'])) {
    $_SESSION['msg'] = insert_question($_POST);
}

if (isset($_POST['addquestfile'])) {
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Добавить вопрос</h2>
<label>Группа специальностей</label>
<select class="form-control" name="group" id="group">
    <option disabled selected>Выберите группу</option>
    <?php foreach($groups as $group): ?>
        <option value="<?=$group['id']?>"><?=$group['group']?></option>
    <?php endforeach; ?>
</select>
<label>Специальность</label>
<select class="form-control" name="special" id="special">
    <option disabled selected>Выберите специальность</option>
</select>

<br>
<br>
```

```

<div class="addquest">
  <input type="text" style="width: 93%; margin-right: 13;" name="question"
placeholder="Введите вопрос">
  <?php for ($i=1; $i < 7; $i++) : ?>
    <input type="text" name="answer-<?=$i?>" placeholder="Введите вариант ответа
<?=$i?>">
    <input type="checkbox" style="width: auto;" name="correct-answer-<?=$i?>">
  <?php endfor; ?>
</div>
<br>
<br>

<input type="submit" name="addquest" value="Добавить">
<input type="submit" name="addquestfile" value="Добавить из файла">

</form>

<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ Е

Листинг изменения вопросов файла adminchangequest.php

```
<?php
session_start();
ini_set("display_errors", 1);
error_reporting(-1);
require_once 'config.php';
require_once 'functions.php';
if (!check_user()) {
    header("Location: index.php");
    exit();
}
if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}
$spec = get_spec();
$categ = get_category();
$id_quest = $_SESSION['change_quest'];
$tab = get_quest($id_quest);
//print_arr($_POST); print_arr($tab); print_arr($_SESSION);exit();
if (isset($_POST['change_quest'])) {
    $msg = change_quest($_POST, $tab);
    if ($msg === TRUE) {
        $_SESSION['msg'] = "Данные пользователя изменены";
        header("Location: adminquest.php");
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Изменить вопрос</h2>
<br>
<div class="addquest">
<?php foreach ($tab as $k => $v) : ?>
    <?php if (!$k) : ?>
        <input type="text" name="<?=$k?>" value="<?=$v?>"> <br>
    <?php else : ?>
        <input type="text" style="margin-left: 3%; width: 87%" name="<?=$k?>" value="<?=$v?>">
    <br>
<?php endif; ?>
<?php endforeach; ?>
<br>
<input type="submit" style="width: 180px" name="change_quest" value="Сохранить">
<br>
</div>
</form>
<?php require_once 'footer.php'; ?>
```

ПРИЛОЖЕНИЕ И

Листинг	изменения	данных	пользователя	файла
---------	-----------	--------	--------------	-------

adminchangeuser.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1)) {
    header("Location: kabinet.php");
    exit();
}

$spec = get_spec();
$mf = get_facility();
$categ = get_category();
$stab = get_user($_SESSION);

if (isset($_POST['change'])) {
    $msg = change_user($_POST, $stab);
    if ($msg === TRUE) {
        $_SESSION['msg'] = "Данные пользователя изменены";
        header("Location: adminusers.php");
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Изменить данные пользователя</h2>
<br>
<div class="changeuser">
    <label>Фамилия</label>
    <input type="text" name="surname" value="<?=$stab['surname']?>"> <br>
    <label>Имя</label>
    <input type="text" name="name" value="<?=$stab['name']?>"> <br>
    <label>Отчество</label>
```

```

        <input type="text" name="patronymic" value="<?=$tab['patronymic']?>"> <br>
        <label>Специальность</label>
        <select class="form-control" name="specialty" id="special">
            <?php foreach($spec as $special): ?>
                <?php if ($special['id'] == $tab['specialty']) : ?>
                    <option selected
value="<?=$special['id']?>"><?=$special['specialty']?></option>
                <?php else : ?>
                    <option
value="<?=$special['id']?>"><?=$special['specialty']?></option>
                <?php endif; ?>
            <?php endforeach; ?>
        </select> <br>
        <label>Стаж</label>
        <input type="text" name="experience" value="<?=$tab['experience']?>"> <br>
        <label>Место работы</label>
        <select class="facility" name="mf" required>
            <option disabled selected value="0">Выберите место работы</option>
            <?php foreach ($mf as $m): ?>
                <?php if ($m['id'] == $tab['mf']) : ?>
                    <option selected value="<?=$m['id']?>"><?=$m['med_facility']?></option>
                <?php else : ?>
                    <option value="<?=$m['id']?>"><?=$m['med_facility']?></option>
                <?php endif; ?>
            <?php endforeach; ?>
        </select> <br>
        <label>Категория</label>
        <select class="category" name="category" required>
            <option disabled selected value="0">Выберите категорию</option>
            <?php foreach ($categ as $category): ?>
                <?php if ($category['id'] == $tab['category']) : ?>
                    <option selected
value="<?=$category['id']?>"><?=$category['category']?></option>
                <?php else : ?>
                    <option
value="<?=$category['id']?>"><?=$category['category']?></option>
                <?php endif; ?>
            <?php endforeach; ?>
        </select> <br>
        <label>Телефон</label>
        <input type="text" maxlength="10" name="phone" value="<?=$tab['phone']?>"> <br>
        <label>Пробный тест</label>
        <input type="text" name="probe_test" value="<?=$tab['probe_test']?>"> <br>
        <label>Основной тест</label>
        <input type="text" name="main_test" value="<?=$tab['main_test']?>"> <br>
        <input type="submit" style="width: 180px" name="change" value="Сохранить"> <br>
    </div>
</form>
<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ К

Листинг изменения отчетов файла adminreports.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';
/* require_once('Classes/PHPExcel.php'); */
// Подключаем класс для вывода данных в формате excel
//require_once('PHPExcel/Writer/Excel5.php');

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$sum = 0;
$sum1 = 0;
$sum2 = 0;
$spec = get_spec();
$mfac = get_facility();
$categ = get_category();
$groups = get_group();

/* if (isset($_POST['export'])) {
    $slena = get_report_excel($_POST, $tab);
} */

if (isset($_POST['view'])) {
    $tab = get_report($_POST);
}

if (isset($tab)) {
    foreach ($tab as $k => $v) {
        foreach ($spec as $s) {
            if ($s['id'] == $v['specialty_t']) {
                $tab[$k]['specialty_t'] = $s['specialty'];
            }
        }
    }
}
```



```

foreach ($tab as $k => $v) {
    foreach ($mfac as $m) {
        if ($m['id'] == $v['mf']) {
            $tab[$k]['mf'] = $m['med_facility'];
        }
    }
}

foreach ($tab as $k => $v) {
    foreach ($categ as $category) {
        if ($category['id'] == $v['category_t']) {
            $tab[$k]['category_t'] = $category['category'];
        }
    }
}

foreach ($tab as $k => $v) {
    if ($v['type_test'] == 'main'){
        $tab[$k]['type_test'] = 'Основной';
    }
    elseif ($v['type_test'] == 'probe') {
        $tab[$k]['type_test'] = 'Пробный';
    }
    elseif ($v['type_test'] == 'additional') {
        $tab[$k]['type_test'] = 'Дополнительная попытка';
    }
}

if (isset($_POST['specperc'])) {
    $tr = all_result();
    $count = array_fill(0, 50, '0');
    $count1 = array_fill(0, 50, '0');
    foreach ($tr as $key => $value) {
        foreach ($spec as $k => $v) {
            if ($value['specialty_t'] == $v['id']){
                if ($value['percent'] >= 70) {
                    $count[$k]++;
                }
                else {
                    $count1[$k]++;
                }
            }
        }
    }
    $j = 0;
    foreach ($spec as $kl => $zn) {
        if ((isset($count[$kl])) || (isset($count1[$kl]))) {
            $diagramm[$j]['specialty'] = $zn['specialty'];
            if (isset($count[$kl])){
                $diagramm[$j]['correct'] = $count[$kl];
            }
        }
    }
}

```

```

        if (isset($count1[$kl])) {
            $diagramm[$j]['incorrect'] = $count1[$kl];
        }
        $j++;
    }

}

foreach ($diagramm as $key => $value) {
    if (($value['correct'] == 0) && ($value['incorrect'] == 0) {
        unset($diagramm[$key]);
    }
}

}

if (isset($_POST['mfperc'])) {
    $tr = all_result();
    $count = array_fill(0, 50, '0');
    $count1 = array_fill(0, 50, '0');
    foreach ($tr as $key => $value) {
        foreach ($mfac as $k => $v) {
            if ($value['mf'] == $v['id']){
                if ($value['percent'] >= 70) {
                    $count[$k]++;
                }
                else {
                    $count1[$k]++;
                }
            }
        }
    }
}

$j = 0;
foreach ($mfac as $kl => $zn) {

    if ((isset($count[$kl])) || (isset($count1[$kl]))) {
        $diagramm[$j]['mfac'] = $zn['med_facility'];
        if (isset($count[$kl])){
            $diagramm[$j]['correct'] = $count[$kl];
        }
        if (isset($count1[$kl])) {
            $diagramm[$j]['incorrect'] = $count1[$kl];
        }
        $j++;
    }

}

foreach ($diagramm as $key => $value) {
    if (($value['correct'] == 0) && ($value['incorrect'] == 0) {
        unset($diagramm[$key]);
    }
}

}

```

```

?>
<?php require_once 'header.php'; ?>
<form method="POST" class="report">
<h2>Отчеты</h2>
<h3> Виды отчетов </h3>

<div class="reportsub1">
<input type="submit" name="specregp" value="Прошедшие тестирование, по
специальностям">
<input type="submit" name="specregn" value="Не прошедшие тестирование, по
специальностям">
<input type="submit" name="specperc" value="Проходимость по специальностям">

</div>
<div class="reportsub2">
<input type="submit" name="mfregp" value="Прошедшие тестирование, по МУ">
<input type="submit" name="mfregn" value="Не прошедшие тестирование, по МУ">
<input type="submit" name="mfperc" value="Проходимость по МУ">
</div>

<div class="reportsub4">
es
</div>

<!--*****Прошедшие /Не прошедшие тестирование, по спец-ти и
датам*****-->
<?php if ((isset($_POST['specregp'])) || (isset($_POST['specregn']))) : ?>
<?php if (isset($_POST['specregp'])) : ?>
<input type="hidden" name="resrt" value="1">
<input type="hidden" name="trep" value="<?=$_POST['specregp']?>">
<?php endif;?>
<?php if (isset($_POST['specregn'])) : ?>
<input type="hidden" name="resrt" value="0">
<input type="hidden" name="trep" value="<?=$_POST['specregn']?>">
<?php endif;?>

<h3>Выберите параметры Отчета</h3>
<select name="resgroup" id="group">
<option disabled selected>Группа специальности, по которой проводилось
тестирование</option>
<?php foreach($groups as $group): ?>
<option value="<?=$group['id']?>"><?=$group['grouppp']?></option>
<?php endforeach; ?>
</select>
<select name="resspecial" id="special">
<option disabled selected>Специальность, по которой проводилось
тестирование</option>
<option disabled>Выберите группу</option>
</select>

<br> <br>

```

```

<label>Дата тестирования С:</label>
<input type="date" style="width: 145px" name="sdate" placeholder="Начальная дата">
<label>По:</label>
<input type="date" style="width: 145px" name="edate" placeholder="Начальная дата">

<br> <br>

<input type="submit" style="width: 20%" name="view" value="Сформировать отчет">
<br> <br>
<?php endif; ?>
<?php if (isset($stab)) : ?>
<?php if (isset($_POST['resspecial'])) : ?>
    <h3> Отчет по специальности </h3>
    <h2> <?=$_POST['trep']?> </h2>
<?php foreach ($spec as $key => $value):?>
    <?php if ($value['id'] == $_POST['resspecial']): ?>
        <?php $spec = $value['specialty']; ?>
    <?php endif; ?>
<?php endforeach;?>

    <h2> Специальность: <?=$spec?> </h2>
<?php endif; ?>

<?php endif; ?>
<!--*****Прошедшие /Не прошедшие тестирование, по МУ и
датам*****-->

<?php if ((isset($_POST['mfregp'])) || (isset($_POST['mfregn']))) : ?>
<?php if (isset($_POST['mfregp'])) : ?>
<input type="hidden" name="resrt" value="1">
<input type="hidden" name="trep" value="<?=$_POST['mfregp']?>">
<?php endif;?>
<?php if (isset($_POST['mfregn'])) : ?>
<input type="hidden" name="resrt" value="0">
<input type="hidden" name="trep" value="<?=$_POST['mfregn']?>">
<?php endif;?>
    <h3>Выберите параметры Отчета</h3>
<select name="resmf" id="mf">
    <option disabled selected value="0">Место работы тестируемого</option>
    <?php foreach ($mfac as $m): ?>
        <?php if ($m['id'] == $stab['mf']) : ?>
            <option selected
value="<?=$m['id']?>"><?=$m['med_facility']?></option>
        <?php else : ?>
            <option value="<?=$m['id']?>"><?=$m['med_facility']?></option>
        <?php endif; ?>
    <?php endforeach; ?>
</select>

<br> <br>
<label>Дата тестирования С:</label>

```

```

<input type="date" style="width: 145px" name="sdate" placeholder="Начальная дата">
<label>По:</label>
<input type="date" style="width: 145px" name="edate" placeholder="Начальная дата">
<br> <br>
<input type="submit" style="width: 20%" name="view" value="Сформировать отчет">
<br> <br>
<?php endif; ?>

```

```

<?php if (isset($tab)) : ?>
<?php if (isset($_POST['resmf'])) : ?>
<h3> Отчет по медицинским учреждениям </h3>
<h2><?=$_POST['trep']?> </h2>
<?php foreach ($mfac as $k => $v):?>
    <?php if ($v['id'] == $_POST['resmf']): ?>
        <?php $mfac = $v['med_facility']; ?>
    <?php endif; ?>
<?php endforeach;?>
<h2>Медицинское учреждение: <?=$mfac;?></h2>
<?php endif; ?> <br>
<?php if (!empty($_POST['sdate'])) : ?>
<h2>С: <?=$_POST['sdate']?></h2>
<?php endif; ?>
<?php if (!empty($_POST['edate'])) : ?>
<h2>По: <?=$_POST['edate']?></h2> <br>
<?php endif; ?>
<?php endif; ?>

```

```

<?php if (isset($tab)) : ?>

<div id="resulttest">
<table class="resulttest">
    <tr>
        <td style="border: none;"></td>
        <th>Всего записей: </th>
        <th><?= count($tab); ?></th>
    </tr>
    <tr style="font-weight: bold;">
        <td>ID</td>
        <td>Дата тестирования</td>
        <td>Тип теста</td>
        <td>Результат</td>
        <td>Фамилия</td>
        <td>Имя</td>
        <td>Отчество</td>
        <td>e-mail</td>
        <td>Телефон</td>
        <td>Специальность</td>
        <td>Категория</td>
        <td>Место работы</td>
    </tr>

```

```

<?php foreach ($stab as $table): ?>

    <tr>
        <td><?=$table['id']?></td>
        <td><?=$table['date']?></td>
        <td><?=$table['type_test']?></td>
        <td><?=$table['result'] . ' ('.$table['percent'] . '%)'?></td>
        <td><?=$table['surname']?></td>
        <td><?=$table['name']?></td>
        <td><?=$table['patronymic']?></td>
        <td><?=$table['email']?></td>
        <td><?=$table['phone']?></td>
        <td><?=$table['specialty_t']?></td>
        <td><?=$table['category_t']?></td>
        <td><?=$table['mf']?></td>
    </tr>
<?php endforeach ?>
</table>
</div>

<?php endif; ?>

<!--*****Таблица проходимости по специальностям*****-->
<?php if (isset($_POST['спесперс'])) : ?>
<h3> Отчет по количеству прошедших и не прошедших тестирование, по
специальностям </h3>
<div id="resulttest">

<table class="resulttest">
    <tr style="font-weight: bold;">
        <td>Специальность</td>
        <td>Количество прошедших</td>
        <td>Количество не прошедших</td>
        <td>Всего</td>
    </tr>
    <?php foreach ($diagramm as $table): ?>
        <tr>
            <td><?=$table['specialty']?></td>
            <td><?=$table['correct']?></td> <?php $sum +=
$stable['correct']; ?>
            <td><?=$table['incorrect']?></td> <?php $sum1 +=
$stable['incorrect']; ?>
            <td style="font-weight: bold;"><?=$table['correct'] +
$stable['incorrect']?></td> <?php $sum2 += ($stable['correct'] + $stable['incorrect']); ?>
        </tr>
    <?php endforeach ?>
    <tr style="font-weight: bold;">
        <td>Всего:</td>
        <td><?=$sum?></td>
        <td><?=$sum1?></td>
        <td><?=$sum2?></td>
    </tr>
</div>

```

```

</table>
</div>
<?php endif;?>
<!--*****Таблица проходимости по МУ*****-->
<?php if (isset($_POST['mfperc'])) : ?>
<h3> Отчет по количеству прошедших и не прошедших тестирование, по
медицинским учреждениям </h3>
<div id="resulttest">

<table class="resulttest">
<tr style="font-weight: bold;">
<td>Медицинское учреждение</td>
<td>Количество прошедших</td>
<td>Количество не прошедших</td>
<td>Всего</td>

</tr>
<?php foreach ($diagramm as $table): ?>
<tr>
<td><?=$table['mfac']?></td>
<td><?=$table['correct']?></td> <?php $sum +=
$table['correct']; ?>
<td><?=$table['incorrect']?></td> <?php $sum1 +=
$table['incorrect']; ?>
<td style="font-weight: bold;"><?=$table['correct'] +
$table['incorrect']?></td> <?php $sum2 += ($table['correct'] + $table['incorrect']); ?>
</tr>
<?php endforeach ?>
<tr style="font-weight: bold;">
<td>Всего:</td>
<td><?=$sum?></td>
<td><?=$sum1?></td>
<td><?=$sum2?></td>

</tr>
</table>
</div>
<?php endif;?>
</form>

<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ Л

Листинг добавления справочников файла adminbook.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$spec = get_spec();
$mf = get_facility();
$group = get_group();

foreach ($spec as $k => $v) {
    foreach ($group as $s) {
        if ($s['id'] == $v['parent_group']) {
            $spec[$k]['parent_group'] = $s['group'];
        }
    }
}

if (isset($_POST['addg'])) {
    header("Location: adminaddgroup.php");
}

if (isset($_POST['unsetg'])) {
    $msg = unset_g($_POST);
    header("Location: ".$_SERVER['PHP_SELF']);
}

if (isset($_POST['addmedfacility'])) {
    header("Location: adminaddmedfac.php");
}

if (isset($_POST['unsetmf'])) {
    $msg = unset_mf($_POST);
    header("Location: ".$_SERVER['PHP_SELF']);
}
```



```

if (isset($_POST['addspecialty'])) {
    header("Location: adminaddspecialty.php");
}
if (isset($_POST['unsetspec'])) {
    $msg = unset_spec($_POST);
    header("Location: ".$_SERVER['PHP_SELF']);
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">

<h2>Справочники</h2>
    <h3> Медицинские учреждения</h3>
    <input type="submit" name="addmedfacility" value="Добавить"> <br>
    <div id="resulttest">
        <table class="resulttest">
            <tr style="font-weight: bold;">
                <td>ID</td>
                <td>Медицинские учреждения</td>
                <td></td>
            </tr>
            <?php foreach ($mf as $table): ?>
                <tr>
                    <td><?=$table['id']?></td>
                    <td><?=$table['med_facility']?></td>
                    <td><button value="<?=$table['id']?>"
name="unsetmf">Удалить</button></td>
                </tr>
            <?php endforeach ?>
        </table>
    </div>
    <br>

    <h3> Группы специальностей</h3>
    <input type="submit" name="addg" value="Добавить"> <br>
    <div id="spravochnik">
        <table class="resulttest">
            <tr style="font-weight: bold;">
                <td>ID</td>
                <td>Группа специальностей</td>
                <td></td>
            </tr>
            <?php foreach ($group as $table): ?>
                <tr>
                    <td><?=$table['id']?></td>
                    <td><?=$table['group']?></td>
                    <td><button value="<?=$table['id']?>"
name="unsetg">Удалить</button></td>
                </tr>
            <?php endforeach ?>
        </table>
    </div>

```

```

<br>

<h3> Специальности</h3>
<input type="submit" name="addspecialty" value="Добавить"> <br>
<div id="resulttest">
<table class="resulttest">
    <tr style="font-weight: bold;">
        <td>ID</td>
        <td>Специальность</td>
        <td>Группа специальностей</td>
        <td></td>
    </tr>
    </tr>
    <?php foreach ($spec as $table): ?>
        <tr>
            <td><?=$table['id']?></td>
            <td><?=$table['specialty']?></td>
            <td><?=$table['parent_group']?></td>
            <td><button                                value="<?=$table['id']?>"
name="unsetspec">Удалить</button></td>
        </tr>
    </tr>
    <?php endforeach ?>
</table>
</div>

</form>

<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ М

Листинг добавления в справочник «Медицинские учреждения» файла adminaddmedfac.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

if (isset($_POST['medfacility'])) {
    $_SESSION['msg'] = insert_medfac($_POST);
}

?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Добавить медицинское учреждение</h2>
<br>
<div class="spravochnik">
    <input type="text" style="width: 80%;" name="medfacility" placeholder="Введите
медицинское учреждение">
    <br>
    <br>
    <input type="submit" name="addmedfacility" value="Добавить">
</div>
</form>
<?php require_once 'footer.php'; ?>
```

ПРИЛОЖЕНИЕ Н

Листинг добавления в справочник «Группы специальностей» файла adminaddgroup.php

```
?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

if (isset($_POST['addgroup'])) {
    $_SESSION['msg'] = insert_group($_POST);
}

?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Добавить группу специальностей</h2>
<br>
<div class="spravochnik">
<input type="text" style="width: 80%;" name="group" placeholder="Введите
группу">
<br>
<br>
<input type="submit" name="addgroup" value="Добавить">
</div>
</form>
<?php require_once 'footer.php'; ?>
```

ПРИЛОЖЕНИЕ П

Листинг добавления в справочник «Специальности» файла adminaddspecialty.php

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}
$groups = get_group();

if (isset($_POST['specialty'])) {
    $_SESSION['msg'] = insert_specialty($_POST);
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Добавить специальность</h2>
<div class="spravochnik">
<select class="form-control" name="group" id="group">
    <option disabled selected>Выберите группу специальностей</option>
    <?php foreach($groups as $group): ?>
        <option value="<?=$group['id']?>"><?=$group['group']?></option>
    <?php endforeach; ?>
</select>
<br>
<br>
<input type="text" style="width: 80%;" name="specialty" placeholder="Введите
специальность">
<br>
<br>
<input type="submit" name="addspecialty" value="Добавить">
</div>
</form>
<?php require_once 'footer.php'; ?>
```

ПРИЛОЖЕНИЕ Р

Листинг вывода списка ожидаемых пользователей файла **adminexpected.php**

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$mf = get_facility();
$spec = get_spec();

$tab = all_expected();

if ($tab){
    foreach ($tab as $k => $v) {
        foreach ($mf as $m) {
            if ($m['id'] == $v['mf']) {
                $tab[$k]['mf'] = $m['med_facility'];
            }
        }
    }
    foreach ($tab as $k => $v) {
        foreach ($spec as $s) {
            if ($s['id'] == $v['specialty']) {
                $tab[$k]['specialty'] = $s['specialty'];
            }
        }
    }
}

if (isset($_POST['unset'])) {
    $msg = unset_expected($_POST);

    if ($msg === TRUE) {
        $_SESSION['msg'] = "Пользователь удален";
    }
}
```

```

        header("Location:".$_SERVER['PHP_SELF']);
    }
    else {
        $_SESSION['msg'] = $msg;
    }
}

if (isset($_POST['addexp'])) {
    header("Location: adminaddexpected.php");
}
?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Список ожидаемых пользователей для тестирования</h2> <br>
<input type="submit" name="addexp" value="Добавить пользователя"> <br> <br>
<?php if ($tab): ?>
<div id="resulttest">
    <table class="resulttest">
        <tr style="font-weight: bold;">
            <td></td>
            <td>ID</td>
            <td>Фамилия</td>
            <td>Имя</td>
            <td>Отчество</td>
            <td>Место работы</td>
            <td>Специальность</td>
            <td>Дата</td>
        </tr>
        <?php foreach ($tab as $table): ?>
        <tr>
            <td><button value="<?=$table['id']?>"
name="unset">Удалить</button></td>
            <td><?=$table['id']?></td>
            <td><?=$table['surname']?></td>
            <td><?=$table['name']?></td>
            <td><?=$table['patronymic']?></td>
            <td><?=$table['mf']?></td>
            <td><?=$table['specialty']?></td>
            <td><?=$table['date']?></td>
        </tr>
        <?php endforeach ?>
    </table>
</div>
<?php endif; ?>
</form>

<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ С

Листинг добавления пользователей в список ожидаемых файла **adminaddexpected.php**

```
<?php
session_start();

ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!check_user()) {
    header("Location: index.php");
    exit();
}

if (isset($_SESSION['a']) && ($_SESSION['a'] != 1) && ($_SESSION['a'] != 2)) {
    header("Location: kabinet.php");
    exit();
}

$mf = get_facility();
$groups = get_group();

if (isset($_POST['addexp'])) {
    $_SESSION['msg'] = insert_expected($_POST);
}

?>
<?php require_once 'header.php'; ?>
<form method="POST">
<h2>Добавить пользователя</h2>

<br>
<div class="addquest" style="border: none;">
<input type="text" name="expsurname" placeholder="Введите Фамилию
пользователя"><br>
<input type="text" name="expname" placeholder="Введите Имя пользователя"><br>
<input type="text" name="exp patronymic" placeholder="Введите Отчество
пользователя"><br>

<select name="mf" id="mf">
    <option disabled selected>Выберите место работы пользователя</option>
    <?php foreach($mf as $mf): ?>
        <option value="<?=$mf['id']?>"><?=$mf['med_facility']?></option>
    <?php endforeach; ?>
</select>
```



```

<select class="form-control" name="group" id="group">
  <option disabled selected>Выберите группу специальностей</option>
  <?php foreach($groups as $group): ?>
    <option value="<?=$group['id']?>"><?=$group['group']?></option>
  <?php endforeach; ?>
</select>

<select class="form-control" name="special" id="special">
  <option disabled selected>Выберите специальность</option>
</select>

</div>
<br>
<input type="submit" style="margin-left: 40%" name="addexp" value="Добавить">

</form>
<?php require_once 'footer.php'; ?>

```

ПРИЛОЖЕНИЕ Т

Листинг файла header.php

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
<script>
    $(document).ready(function(){
        $('#login-trigger').click(function(){
            $(this).next('#login-content').slideToggle();
        });
    });
</script>

<?php
ini_set("display_errors", 1);
error_reporting(-1);

require_once 'config.php';
require_once 'functions.php';

if (!isset($_SESSION['msg'])) {
    $_SESSION['msg'] = "";
}

if (isset($_POST['login']) && isset($_POST['password'])) {
    $msg = login($_POST);
    if ($msg === TRUE) {
        if (($_SESSION['a'] == 1) or ($_SESSION['a'] == 2)) {
            header("Location: adminresult.php");
        }
        else {
            header("Location: kabinet.php");
        }
    }
    else {
        $_SESSION['msg'] = $msg;
        header("Location: ".$_SERVER['PHP_SELF']);
    }
    exit();
}

if (isset($_POST['logout'])) {
    $msg = logout();
    if ($msg === TRUE) {
        header("Location: ".$_SERVER['PHP_SELF']);
    }
    exit();
}
```

```

}
?>

<html>
<head>
<title> TMC </title>
<link rel="shortcut icon" href="images//siteicon.ico" type="image/x-icon">
<meta charset="utf-8">
    <link rel="stylesheet" type="text/css" href="fff.css">
</head>
<body>
<div class="page">
<div class="header">
    <div class="logo">
        <a href="index.php"></a> <h1>
Система тестирования медицинских работников </h1>
    </div>
    <?php if (!isset($_SESSION['sess'])): // ВЫВОДИМ ОКНО ВХОДА ?>
    <div class="searchform">
    <nav>
    <ul>
        <li id="login">
            <a id="login-trigger" href="#">
                Войти <span>&#x25BC;</span>
            </a>
            <div id="login-content">
                <form method="POST" action="" style="margin: 0px">
                <input id="username" type="text" name="login" placeholder="Логин" required>
                <input id="password" type="password" name="password" placeholder="Пароль"
required>
                <label><input type="checkbox" name="member" value="1" style="width: 20px;">
Запомнить меня</label>
                <input type="submit" style="width: 110px;" value="Войти">
                </form>
                <a href="returnpass.php"><label>Забыли пароль?</label></a>
            </div> <!--id="login-content"-->
        </li>
        <li id="signup">
            <a href="register.php">Регистрация</a>
        </li>
    </ul>
    </nav>
    </div> <!--class="searchform"-->
    <?php else: //ВЫВОДИМ ОКНО ВЫХОДА ?>
    <div class="searchform">
        <div id="login-content">
            <h1>Добро пожаловать, <a href="kabinet.php">
<?=$_SESSION['name']?> <?=$_SESSION['patronymic']?>!</a></h1>
            </div> <!--id="login-content"-->
        </div> <!--class="searchform"-->
    <?php endif;?>
    </div> <!--class="header"-->

```

```

<form style="margin-bottom: 0px;" method="POST">
<div class="menu-main">
    <ul class="menu-main">
        <?php if (isset($_SESSION['a']) && isset($_SESSION['sess']) && (($_SESSION['a'] ==
1) or ($_SESSION['a'] == 2))): ?>
            <li><a href="adminresult.php">Результаты тестирований</a></li>
            <li><a href="adminusers.php">Пользователи</a></li>
            <li><a href="adminquest.php">Банк вопросов</a></li>
            <li><a href="adminaddquest.php">Добавить вопрос</a></li>
            <?php if (isset($_SESSION['a']) && ($_SESSION['a'] == 1) &&
isset($_SESSION['sess'])): ?>
                <li><a href="adminaddquest.php">Справочники</a></li>
                <?php endif; ?>
                <li><a href="adminreports.php">Отчеты</a></li>
            <?php else: ?>
                <li><a href="index.php">Главная</a></li>
            <?php if (isset($_SESSION['sess'])): ?>
                <li><a href="kabinet.php">Личный кабинет</a></li>
                <li><a href="testp.php">Пройти тестирование</a></li>
                <li><a href="kabinetresult.php">Результаты тестирований</a></li>
                <?php endif; ?>
            <li><a href="reference.php">Справка</a></li>
            <?php endif; ?>
            <?php if (isset($_SESSION['sess'])): ?>
            <li><a><input type="submit" id="logout" name="logout" value="Выйти"></a></li>
            <?php endif; ?>
        </ul>
    </div> <!--class="menu-main"-->
</form>
<div class="errors">
    <?php if (!empty($_SESSION['msg'])): ?>
        <br>
        <?=$_SESSION['msg']?>
        <?php unset($_SESSION['msg'])?>
        <br>
    <?php endif; ?>
</div> <!--class="errors"-->

<div class="content">

```

ПРИЛОЖЕНИЕ У

Листинг обработки функций файла «functions.php»

```
/**
 * Получить всех пользователей
 **/
function all_users() {
    global $db;
    $query = "SELECT *
              FROM users";
    $result = mysqli_query($db, $query);
    while ($row = mysqli_fetch_assoc($result)) {
        $data[] = $row;
    }
    $data = array_reverse($data, true);

    return $data;
}

function insert_question($POST) {
    if (!$POST) return;
    global $db;

    if ($POST['question'] == NULL) {
        return "Введите вопрос";
    }

    if (!isset($POST['special'])) {
        return "Выберите специальность";
    }

    $question = $POST['question'];
    for ($i=1; $i < 7; $i++) {
        if ($POST['answer-' . $i] != NULL) {
            $answer[$i] = $POST['answer-' . $i];
            if ((isset($POST['correct-answer-' . $i])) && ($POST['correct-answer-' . $i]
== 'on')) {
                $correct_answer[$i] = 1;
            }
            else {
                $correct_answer[$i] = 0;
            }
        }
    }
    $category = 1;
    $specialty = $POST['special'];

    if (!isset($answer)) {
        return "Введите варианты ответов";
    }
}
```

```

        if (!isset($correct_answer)) {
            return "Выберите верный ответ";
        }

        $query = "INSERT INTO questions (question, category, specialty)
                VALUES ('$question','$category','$specialty')";
        $result = mysqli_query($db, $query);

        if (!$result) {
            return "Ошибка при добавлении вопросов в базу данных";
        }
        else {

            $sql = "SELECT id
                    FROM questions
                    WHERE question = '$question'";
            $result2 = mysqli_query($db, $sql);
            $id_quest = mysqli_fetch_assoc($result2);
            $id_quest = $id_quest['id'];
            foreach ($answer as $k => $ans) {

                $query = "INSERT INTO answers (answer, parent_question, correct_answer)
                        VALUES ('$ans','$id_quest','$correct_answer[$k]')";
                $result3 = mysqli_query($db, $query);

            }

            if (!$result3) {
                return "Ошибка при добавлении ответов в базу данных";
            }
            else {

                $query = "INSERT INTO questions_specialties (answer, parent_question,
correct_answer)
                        VALUES ('$ans','$id_quest','$correct_answer[$k]')";
                $result3 = mysqli_query($db, $query);
                return "Вопрос и ответы успешно добавлены";
            }
        }
    }

    function insert_specialty($POST) {
        if (!$POST) return;
        global $db;

        if ($POST['specialty'] == NULL) {
            return "Введите наименование специальности";
        }

        if (!isset($POST['group'])) {
            return "Выберите группу";
        }
    }

```

```

$group = $POST['group'];
$specialty = $POST['specialty'];

$query = "INSERT INTO specialties (specialty, parent_group)
VALUES ('$specialty','$group')";
$result = mysqli_query($db, $query);

if (!$result) {
    return "Ошибка при добавлении в базу данных";
}

else {
    return "Специальность успешно добавлена";
}
}

function insert_medfac($POST) {
    if (!$POST) return;
    global $db;

    if ($POST['medfacility'] == NULL) {
        return "Введите наименование медицинского учреждения";
    }

    $med_facility = $POST['medfacility'];

    $query = "INSERT INTO medical_facility (med_facility)
VALUES ('$med_facility')";
    $result = mysqli_query($db, $query);

    if (!$result) {
        return "Ошибка при добавлении в базу данных";
    }

    else {
        return "Медицинское учреждение успешно добавлено";
    }
}

function insert_group($POST) {
    if (!$POST) return;
    global $db;

    if ($POST['group'] == NULL) {
        return "Введите наименование группы";
    }

    $group = $POST['group'];

    $query = "INSERT INTO groups (group)
VALUES ('$group')";
    $result = mysqli_query($db, $query);

```

```

        if (!$result) {
            return "Ошибка при добавлении группы в базу данных";
        }
        else {
            return "Группа успешно добавлена";
        }
    }

function block_user($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_user = $POST['block'];

    $sql = "UPDATE users
            SET block = '1'
            WHERE user_id = '$id_user'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно заблокировать пользователя. Ошибка при
обращении к БД.";
    }

    return TRUE;
}

function unblock_user($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_user = $POST['unblock'];

    $sql = "UPDATE users
            SET block = '0'
            WHERE user_id = '$id_user'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно заблокировать пользователя. Ошибка при
обращении к БД.";
    }

    return TRUE;
}

function get_user($SESSION) {
    global $db;
    if (!$SESSION) return;

    if (isset($SESSION['change'])) {

```



```

        $id_user = $SESSION['change'];
    }
    elseif (isset($SESSION['user_id'])) {
        $id_user = $SESSION['user_id'];
    }

    $sql = "SELECT *
            FROM users
            WHERE user_id = '$id_user'";
    $result = mysqli_query($db, $sql);

    $data = NULL;
    while ($row = mysqli_fetch_assoc($result)) {
        $data = $row;
    }
    return $data;
}

function change_user($POST, $tab) {
    global $db;

    $id = $tab['user_id'];

    $surname = $POST['surname'];
    $name = $POST['name'];
    $patronymic = $POST['patronymic'];
    $specialty = $POST['specialty'];
    $experience = $POST['experience'];
    $mf = $POST['mf'];
    $category = $POST['category'];
    $phone = $POST['phone'];
    $probe_test = $POST['probe_test'];
    $main_test = $POST['main_test'];

    $sql = "UPDATE users
            SET surname = '$surname',
                name = '$name',
                patronymic = '$patronymic',
                specialty = '$specialty',
                experience = '$experience',
                mf = '$mf',
                category = '$category',
                phone = '$phone',
                probe_test = '$probe_test',
                main_test = '$main_test'
            WHERE user_id = '$id'";

    $result = mysqli_query($db, $sql);

    if (!$result) {

```

```

        return "Невозможно изменить данные пользователя. Ошибка при
обращении к БД.";
    }

    return TRUE;
}

function change_users($post) {
    global $db;

    $id = $_SESSION['user_id'];

    $email = clean_data($post['email']);
    $password = md5(trim($post['password']));
    $conf_password = md5(trim($post['conf_password']));
    $surname = clean_data($post['surname']);
    $name = clean_data($post['name']);
    $patronymic = clean_data($post['patronymic']);
    $specialty = clean_data($post['specialty']);
    $experience = clean_data($post['experience']);
    $mf = clean_data($post['mf']);
    $category = clean_data($post['category']);
    $phone = clean_data($post['phone']);

    $msg = "";
    if ($conf_password == $password) {
        $sql = "UPDATE users
                SET email = '$email',
                    password = '$password',
                    surname = '$surname',
                    name = '$name',
                    patronymic = '$patronymic',
                    specialty = '$specialty',
                    experience = '$experience',
                    mf = '$mf',
                    category = '$category',
                    phone = '$phone'
                WHERE user_id = '$id'";

        $result = mysqli_query($db, $sql);

        if (!$result) {
            return "Невозможно изменить данные пользователя. Ошибка при
обращении к БД.";
        }
        else {
            return "Вы неправильно подтвердили пароль!";
        }

        return TRUE;
    }
}

```

```

function change_quest($POST, $tab) {
    global $db;

    $id = $_SESSION['change_quest'];

    $question = $POST[0];
    unset($POST[0]);
    unset($POST['change_quest']);
    $sql = "UPDATE questions
            SET question = '$question'
            WHERE id = '$id'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно изменить вопрос. Ошибка при обращении к БД.";
    }

    foreach ($POST as $key => $value) {
        $sql1 = "UPDATE answers
                SET answer = '$value'
                WHERE id = '$key'";
        $result1 = mysqli_query($db, $sql1);
        if (!$result1) {
            return "Невозможно изменить ответ. Ошибка при обращении к БД.";
        }
    }

    return TRUE;
}

function unset_quest($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_quest = $POST['unset'];

    $sql = "DELETE FROM questions
            WHERE id = '$id_quest'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно удалить вопрос. Ошибка при обращении к БД.";
    }

    $sql1 = "DELETE FROM answers
            WHERE parent_question = '$id_quest'";
    $result1 = mysqli_query($db, $sql1);

    if (!$result1) {
        return "Невозможно удалить варианты ответов. Ошибка при
обращении к БД.";
    }
}

```

```

    }

    return TRUE;
}

function null_user($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_user = $POST['null'];

    $sql = "UPDATE users
            SET probe_test = '0',
              main_test = '0',
              main_test1 = '-1'
            WHERE user_id = '$id_user'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно обнулить попытки тестирования пользователя.
Ошибка при обращении к БД.";
    }

    return TRUE;
}

function add($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_user = $POST['add'];

    $sql = "UPDATE users
            SET   main_test1 = '0'
            WHERE user_id = '$id_user'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно добавить попытку тестирования пользователю.
Ошибка при обращении к БД.";
    }

    return TRUE;
}

function addg($POST) {
    global $db;
    if (!$POST) return;
    $msg = "";
    $id_user = $POST['add'];

```

```

        $sql = "UPDATE groupp
                WHERE user_id = '$id_    groupp'";
        $result = mysqli_query($db, $sql);

        if (!$result) {
            return "Невозможно добавить попытку тестирования пользователю.
Ошибка при обращении к БД.";
        }

        return TRUE;
    }

function insert_question_file() {
    global $db;

    if (!isset($_POST['special'])) {
        return "Выберите специальность";
    }
    else {
        $specialty = $_POST['special'];
    }

    if (!$_FILES['uploadfile']) {
        return "Выберите файл";
    }
    $file = $_FILES['uploadfile']['tmp_name'];

    $specialty = 2;
    $type_question = 'radio';
    $category = 1;

    $handle = fopen($file, "rt");
    $contents = fread($handle, filesize("probe.qrt"));
    fclose($handle);

    $query = explode("?", $contents);

    foreach ($query as $key => $value) {
        $query[$key] = explode(":", $value);
    }

    foreach ($query as $key => $value) {
        $query[$key][1] = explode("\n", $value[1]);
        foreach ($query[$key][1] as $k => $v) {

            if ($query[$key][1][$k]){
                if ($query[$key][1][$k][0] == '+') {
                    $correct_answer[$key][$k] = '1';
                }
                elseif ($query[$key][1][$k][0] == '-') {
                    $correct_answer[$key][$k] = '0';
                }
            }
        }
    }
}

```

```

        $pos = strpos($v, '.');
        $query[$key][1][$k] = substr($v, $pos+1);
    }
    else {
        unset ($query[$key][1][$k]);
    }
}
$pos1 = strpos($value[0], '.');
$query[$key][0] = substr($value[0], $pos1+1);
}

foreach ($query as $key => $value) {

    $question = $value[0];
    $sql = "INSERT INTO questions (question, type_question, category,
specialty)
        VALUES
('$question','$type_question','$category','$specialty')";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Ошибка при добавлении вопросов в базу данных" ;
    }
    else {
        $sql1 = "SELECT id
                FROM questions
                WHERE question = '$question'";
        $result1 = mysqli_query($db, $sql1);
        $id_quest = mysqli_fetch_assoc($result1);
        $parent_quest = $id_quest['id'];
    }
    foreach ($value[1] as $k => $answer) {
        $correct_answers = $correct_answer[$key][$k];

        $sql2 = "INSERT INTO answers (answer, parent_question,
correct_answer)
        VALUES
('$answer','$parent_quest','$correct_answers')";
        $result2 = mysqli_query($db, $sql2);

        if (!$result2) {
            return "Ошибка при добавлении ответов в базу
данных". $k;
        }
    }
}
return "Вопросы и ответы успешно добавлены";
}

function insert_expected($POST) {
    if (!$POST) return;

```

```

global $db;

if ($POST['expsurname'] == NULL) {
    return "Введите Фамилию";
}
if ($POST['expname'] == NULL) {
    return "Введите Имя";
}
if ($POST['exppatronymic'] == NULL) {
    return "Введите Отчество";
}

/* if ($POST['expdate'] == NULL) {
    return "Введите Дату";
} */
if (!isset($POST['mf'])) {
    return "Выберите Место работы";
}

$surname = $POST['expsurname'];
$name = $POST['expname'];
$patronymic = $POST['exppatronymic'];
$date = date('d-m-Y');
$mf = $POST['mf'];

$query = "INSERT INTO expected (surname, name, patronymic, mf, date)
        VALUES ('$surname','$name','$patronymic','$mf', now())";
$result = mysqli_query($db, $query);

if (!$result) {
    return "Ошибка при добавлении пользователя";
}
else {
    return "Пользователь успешно добавлен";
}
}

function all_expected() {
    global $db;
    $query = "SELECT *
            FROM expected";
    $result = mysqli_query($db, $query);
    if (mysqli_num_rows($result) == 0) {
        $_SESSION['msg'] = "Нет пользователей в базе данных";
        return false;
    }
    while ($row = mysqli_fetch_assoc($result)) {
        $data[] = $row;
    }
    $data = array_reverse($data, true);

    return $data;
}

```

```

    }

    function unset_expected($POST) {
        global $db;
        if (!$POST) return;
        $msg = "";
        $id = $POST['unset'];

        $sql = "DELETE FROM expected
                WHERE id = '$id'";
        $result = mysqli_query($db, $sql);

        if (!$result) {
            return "Невозможно удалить пользователя. Ошибка при обращении к
БД.";
        }
        return TRUE;
    }

    function unset_mf($POST) {
        global $db;
        if (!$POST) return ;
        $msg = "";
        $id = $POST['unsetmf'];

        $sql = "DELETE FROM medical_facility
                WHERE id = '$id'";
        $result = mysqli_query($db, $sql);

        if (!$result) {
            return "Невозможно удалить медучреждение. Ошибка при обращении
к БД.";
        }
        else {
            return "Медучреждение удалено.";
        }
        return TRUE;
    }

    function unset_g($POST) {
        global $db;
        if (!$POST) return ;
        $msg = "";
        $id = $POST['unsetg'];

        $sql = "DELETE FROM groups
                WHERE id = '$id'";
        $result = mysqli_query($db, $sql);

        if (!$result) {
            return "Невозможно удалить группу. Ошибка при обращении к БД.";
        }
    }

```



```

        else {
            return "Группа удалена.";
        }
        return TRUE;
    }

function unset_spec($POST) {
    global $db;
    if (!$POST) return ;
    $msg = "";
    $id = $POST['unsetspec'];

    $sql = "DELETE FROM specialties
            WHERE id = '$id'";
    $result = mysqli_query($db, $sql);

    if (!$result) {
        return "Невозможно удалить специальность. Ошибка при обращении к
БД.";
    }
    else {
        return "Специальность удалена.";
    }
    return TRUE;
}

function get_report ($POST) {
    if (!$POST) return ;
    global $db;

    //print_arr($POST); exit();
    //Прошедшие тестирование, по специальности и датам
    if (isset($POST['resrt']) && $POST['resrt'] == 1) {
        if ((isset($POST['resspecial'])) && (!empty($POST['sdate'])) &&
(!empty($POST['edate']))) {
            if ($POST['sdate'] > $POST['edate']){
                $_SESSION['msg'] = "Неккоректные данные";
                return ;
            }
            $sd = $POST['sdate'];
            $ed = $POST['edate'];
            $spec = $POST['resspecial'];
            $sql = "SELECT *
                    FROM test_results t
                    LEFT JOIN users u
                    ON t.user_id = u.user_id
                    WHERE t.specialty_t = '$spec'
                       AND t.percent >=70
                       AND (t.date BETWEEN '$sd' AND '$ed')";
        }
        elseif ((isset($POST['resspecial'])) && (!empty($POST['sdate']))) {
            if ($POST['sdate'] > date("Y-m-d")){

```

```

        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = date("Y-m-d");
    $spec = $POST['resspecial'];
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.specialty_t = '$spec'
                  AND t.percent >=70
                  AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif ((isset($POST['resspecial'])) && (!empty($POST['edate']))) {
    $sd = '2017-01-01';
    $ed = $POST['edate'];
    $spec = $POST['resspecial'];
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.specialty_t = '$spec'
                  AND t.percent >=70
                  AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif ((!empty($POST['sdate'])) && (!empty($POST['edate']))) {
    if ($POST['sdate'] > $POST['edate']){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = $POST['edate'];
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.percent >=70
                  AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (!empty($POST['sdate'])) {
    if ($POST['sdate'] > date("Y-m-d")){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = date("Y-m-d");
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.percent >=70

```

```

        AND (t.date BETWEEN '$sd' AND '$ed');
    }
    elseif (!empty($POST['edate'])) {
        $sd = '2017-01-01';
        $ed = $POST['edate'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.percent >=70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif (isset($POST['resspecial'])) {
        $spec = $POST['resspecial'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.specialty_t = '$spec'
                AND t.percent >=70";
    }
}

//Не прошедшие тестирование, по специальности и датам
elseif (isset($POST['resrt']) && $POST['resrt'] == 0) {
    if ((isset($POST['resspecial'])) && (!empty($POST['sdate'])) &&
        (!empty($POST['edate']))) {
        if ($POST['sdate'] > $POST['edate']){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = $POST['edate'];
        $spec = $POST['resspecial'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.specialty_t = '$spec'
                AND t.percent < 70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif ((isset($POST['resspecial'])) && (!empty($POST['sdate']))) {
        if ($POST['sdate'] > date("Y-m-d")){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = date("Y-m-d");
        $spec = $POST['resspecial'];
        $sql = "SELECT *
                FROM test_results t

```

```

        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE t.specialty_t = '$spec'
            AND t.percent < 70
            AND (t.date BETWEEN '$sd' AND '$ed')";
    }
elseif ((isset($POST['resspecial'])) && (!empty($POST['edate']))) {
    $sd = '2017-01-01';
    $ed = $POST['edate'];
    $spec = $POST['resspecial'];
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE t.specialty_t = '$spec'
            AND t.percent < 70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif ((!empty($POST['sdate'])) && (!empty($POST['edate']))) {
    if ($POST['sdate'] > $POST['edate']){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = $POST['edate'];
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE t.percent < 70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (!empty($POST['sdate'])) {
    if ($POST['sdate'] > date("Y-m-d")){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = date("Y-m-d");
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE t.percent < 70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (!empty($POST['edate'])) {
    $sd = '2017-01-01';
    $ed = $POST['edate'];
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u

```

```

        ON t.user_id = u.user_id
        WHERE t.percent < 70
            AND (t.date BETWEEN '$sd' AND '$ed')";
    }
elseif (isset($POST['resspecial'])) {
    $spec = $POST['resspecial'];
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE t.specialty_t = '$spec'
            AND t.percent < 70";
    }
}

//*****//
//Прошедшие тестирование, по МУ и датам
if (isset($POST['resrt']) && $POST['resrt'] == 1) {
    if ((isset($POST['resmf'])) && (!empty($POST['sdate'])) &&
(!empty($POST['edate']))) {
        if ($POST['sdate'] > $POST['edate']){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = $POST['edate'];
        $mf = $POST['resmf'];
        $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE u.mf = '$mf'
                AND t.percent >=70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
elseif ((isset($POST['resmf'])) && (!empty($POST['sdate']))) {
    if ($POST['sdate'] > date("Y-m-d")){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = date("Y-m-d");
    $mf = $POST['resmf'];
    $sql = "SELECT *
        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE u.mf = '$mf'
            AND t.percent >=70
            AND (t.date BETWEEN '$sd' AND '$ed')";
    }
elseif ((isset($POST['resmf'])) && (!empty($POST['edate']))) {

```

```

        $sd = '2017-01-01';
        $ed = $POST['edate'];
        $mf = $POST['resmf'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE u.mf = '$mf'
                AND t.percent >=70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
elseif ((!empty($POST['sdate'])) && (!empty($POST['edate']))) {
    if ($POST['sdate'] > $POST['edate']){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = $POST['edate'];
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.percent >=70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (!empty($POST['sdate'])) {
    if ($POST['sdate'] > date("Y-m-d")){
        $_SESSION['msg'] = "Неккоректные данные";
        return ;
    }
    $sd = $POST['sdate'];
    $ed = date("Y-m-d");
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.percent >=70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (!empty($POST['edate'])) {
    $sd = '2017-01-01';
    $ed = $POST['edate'];
    $sql = "SELECT *
            FROM test_results t
            LEFT JOIN users u
            ON t.user_id = u.user_id
            WHERE t.percent >=70
            AND (t.date BETWEEN '$sd' AND '$ed')";
}
elseif (isset($POST['resmf'])) {
    $mf = $POST['resmf'];
    $sql = "SELECT *

```

```

        FROM test_results t
        LEFT JOIN users u
        ON t.user_id = u.user_id
        WHERE u.mf = '$mf'
              AND t.percent >=70";
    }

}
//Не прошедшие тестирование, по МУ и датам
elseif (isset($POST['resrt']) && $POST['resrt'] == 0) {
    if ((isset($POST['resmf'])) && (!empty($POST['sdate'])) &&
(!empty($POST['edate']))) {
        if ($POST['sdate'] > $POST['edate']){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = $POST['edate'];
        $mf = $POST['resmf'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE u.mf = '$mf'
                      AND t.percent < 70
                      AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif ((isset($POST['resmf'])) && (!empty($POST['sdate']))) {
        if ($POST['sdate'] > date("Y-m-d")){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = date("Y-m-d");
        $mf = $POST['resmf'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE u.mf = '$mf'
                      AND t.percent < 70
                      AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif ((isset($POST['resmf'])) && (!empty($POST['edate']))) {
        $sd = '2017-01-01';
        $ed = $POST['edate'];
        $mf = $POST['resmf'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE u.mf = '$mf'

```

```

        AND t.percent < 70
        AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif ((!empty($POST['sdate'])) && (!empty($POST['edate']))) {
        if ($POST['sdate'] > $POST['edate']){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = $POST['edate'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.percent < 70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif (!empty($POST['sdate'])) {
        if ($POST['sdate'] > date("Y-m-d")){
            $_SESSION['msg'] = "Неккоректные данные";
            return ;
        }
        $sd = $POST['sdate'];
        $ed = date("Y-m-d");
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.percent < 70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif (!empty($POST['edate'])) {
        $sd = '2017-01-01';
        $ed = $POST['edate'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE t.percent < 70
                AND (t.date BETWEEN '$sd' AND '$ed')";
    }
    elseif (isset($POST['resmf'])) {
        $mf = $POST['resmf'];
        $sql = "SELECT *
                FROM test_results t
                LEFT JOIN users u
                ON t.user_id = u.user_id
                WHERE u.mf = '$mf'
                AND t.percent < 70";
    }
}

```



```

//*****//
    if (isset($sql)) {
        $result = mysqli_query($db, $sql);
        if (mysqli_num_rows($result) == 0){
            $_SESSION['msg'] = "Данных нет";
            return ;
        }
        while ($row = mysqli_fetch_assoc($result)) {
            $data[] = $row;
        }
    }
    else {
        $_SESSION['msg'] = "Выберите параметры";
        return ;
    }

    //print_arr($POST); print_arr($data); exit();

    return $data;
}

function get_report_excel() {
//Создание объекта класса библиотеки
    $objPHPExcel = new PHPExcel();

    //Указываем страницу, с которой работаем
    $objPHPExcel->setActiveSheetIndex(0);

    //Получаем страницу, с которой будем работать
    $active_sheet = $objPHPExcel->getActiveSheet();

    //Создание новой страницы(пример)
    //$objPHPExcel->createSheet();

    //Ориентация и размер страницы
    // $active_sheet->getPageSetup()
    // -
    >setOrientation(PHPExcel_Worksheet_PageSetup::ORIENTATION_PORTRAIT);
    $active_sheet->getPageSetup()
    -
    >setOrientation(PHPExcel_Worksheet_PageSetup::ORIENTATION_LANDSCAPE);
    $active_sheet->getPageSetup()
    ->setPaperSize(PHPExcel_Worksheet_PageSetup::PAPERSIZE_A4);

    //Имя страницы
    $active_sheet->setTitle("Данные из docs");

    //Ширина столбцов
    $active_sheet->getColumnDimension('A')->setWidth(8);
    $active_sheet->getColumnDimension('B')->setWidth(10);
    $active_sheet->getColumnDimension('C')->setWidth(90);

```

```

$active_sheet->getColumnDimension('D')->setWidth(8);
$active_sheet->getColumnDimension('E')->setWidth(10);
$active_sheet->getColumnDimension('F')->setWidth(90);
    $active_sheet->getColumnDimension('G')->setWidth(8);
$active_sheet->getColumnDimension('H')->setWidth(10);
$active_sheet->getColumnDimension('I')->setWidth(90);
    $active_sheet->getColumnDimension('J')->setWidth(8);
$active_sheet->getColumnDimension('K')->setWidth(10);
$active_sheet->getColumnDimension('L')->setWidth(90); //Объединение ячеек
$active_sheet->mergeCells('A1:C1');

//Высота строки
$active_sheet->getRowDimension('1')->setRowHeight(30);

//Вставить данные(примеры)
//Нумерация строк начинается с 1, координаты A1 - 0,1
$active_sheet->setCellValueByColumnAndRow(0, 1, 'Сегодня '.date('d-m-Y'));
$active_sheet->setCellValue('A3', 'id');
$active_sheet->setCellValue('B3', 'name');
$active_sheet->setCellValue('C3', 'info');
$active_sheet->setCellValue('D3', 'id');
$active_sheet->setCellValue('E3', 'name');
$active_sheet->setCellValue('F3', 'info');
    $active_sheet->setCellValue('G3', 'id');
$active_sheet->setCellValue('H3', 'name');
$active_sheet->setCellValue('I3', 'info');
    $active_sheet->setCellValue('J3', 'id');
$active_sheet->setCellValue('K3', 'name');

//Вставка данных из выборки
$start = 4;
$i = 0;
foreach($l as $row_1){
    $next = $start + $i;

    $active_sheet->setCellValueByColumnAndRow(0, $next, $table['date']);
    $active_sheet->setCellValueByColumnAndRow(1, $next, $table['type_test']);
    $active_sheet->setCellValueByColumnAndRow(2, $next, $row_1['info']);
    $active_sheet->setCellValueByColumnAndRow(3, $next, $table['percent']);
    $active_sheet->setCellValueByColumnAndRow(4, $next, $table['surname']);
    $active_sheet->setCellValueByColumnAndRow(5, $next, $table['name']);
        $active_sheet->setCellValueByColumnAndRow(6, $next,
$stable['patronymic']);
        $active_sheet->setCellValueByColumnAndRow(7, $next, $table['email']);
        $active_sheet->setCellValueByColumnAndRow(8, $next, $table['phone']);
            $active_sheet->setCellValueByColumnAndRow(9, $next,
$stable['specialty_t']);
            $active_sheet->setCellValueByColumnAndRow(10, $next, $table['category_t']);
                $active_sheet->setCellValueByColumnAndRow(11, $next, $table['mf']);
        $i++;
    };
}

```

```

//Отправляем заголовки с типом контекста и именем файла
header("Content-Type:application/vnd.ms-excel");
header("Content-Disposition:attachment;filename='simple.xlsx'");

//Сохраняем файл с помощью PHPExcel_IOFactory и указываем тип Excel
$objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel, 'Excel2007');

//Отправляем файл
$objWriter->save('php://output');

        if (!$objWriter) {
            return "Невозможно удалить специальность. Ошибка при обращении к
БД.";
        }
        else {
            return "Специальность удалена.";
        }
        return TRUE;
    }
}

```

Выпускная квалификационная работа выполнена мной самостоятельно.
Использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

Отпечатано в одном экземпляре.

Библиография _____ наименований.

Один экземпляр сдан на кафедру.

«_____» _____ 2017 г.
(дата)

(подпись)

Е. Е . Султрекова
(ФИО)